

Variable Names – 1

1. C99 standard guarantees uniqueness of _____ characters for internal names.

- a) 31
- b) 63
- c) 12
- d) 14

Answer: b

Explanation: ISO C99 compiler may consider only first 63 characters for internal names.

2. C99 standard guarantees uniqueness of _____ characters for external names.

- a) 31
- b) 6
- c) 12
- d) 14

Answer: a

Explanation: ISO C99 compiler may consider only first 31 characters for external names.

3. Which of the following is not a valid variable name declaration?

- a) int __a3;
- b) int __3a;
- c) int __A3;
- d) None of the mentioned

Answer: d

Explanation: None.

4. Which of the following is not a valid variable name declaration?

- a) int _a3;
- b) int a_3;
- c) int 3_a;
- d) int _3a

Answer: c

Explanation: Variable name cannot start with a digit.

5. Why do variable names beginning with the underscore is not encouraged?

- a) It is not standardized
- b) To avoid conflicts since assemblers and loaders use such names
- c) To avoid conflicts since library routines use such names
- d) To avoid conflicts with environment variables of an operating system

Answer: c

Explanation: None.

6. All keywords in C are in _____

- a) LowerCase letters
- b) UpperCase letters
- c) CamelCase letters
- d) None of the mentioned

Answer: a

Explanation: None.

7. Variable name resolution (number of significant characters for the uniqueness of variable) depends on _____

- a) Compiler and linker implementations
- b) Assemblers and loaders implementations
- c) C language
- d) None of the mentioned

Answer: a

Explanation: It depends on the standard to which compiler and linkers are adhering.

8. Which of the following is not a valid C variable name?

- a) int number;
- b) float rate;
- c) int variable_count;
- d) int \$main;

Answer: d

Explanation: Since only underscore and no other special character is allowed in a variable name, it results in an error.

9. Which of the following is true for variable names in C?

- a) They can contain alphanumeric characters as well as special characters
- b) It is not an error to declare a variable to be one of the keywords (like goto, static)
- c) Variable names cannot start with a digit
- d) Variable can be of any length

Answer: c

Explanation: According to the syntax for C variable name, it cannot start with a digit.

Variable Names – 2

1. Which is valid C expression?

- a) int my_num = 100,000;
- b) int my_num = 100000;
- c) int my num = 1000;
- d) int \$my_num = 10000;

Answer: b

Explanation: Space, comma and \$ cannot be used in a variable name.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("Hello World! %d \n", x);
5.      return 0;
6.  }
```

- a) Hello World! x;
- b) Hello World! followed by a junk value
- c) Compile time error
- d) Hello World!

Answer: c

Explanation: It results in an error since x is used without declaring the variable x.

Output:

\$ cc pgm1.c

pgm1.c: In function 'main':

pgm1.c:4: error: 'x' undeclared (first use in this function)

pgm1.c:4: error: (Each undeclared identifier is reported only once

pgm1.c:4: error: for each function it appears in.)

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int y = 10000;
5.      int y = 34;
6.      printf("Hello World! %d\n", y);
7.      return 0;
8.  }
```

- a) Compile time error
- b) Hello World! 34
- c) Hello World! 1000
- d) Hello World! followed by a junk value

Answer: a

Explanation: Since y is already defined, redefining it results in an error.

Output:

\$ cc pgm2.c

pgm2.c: In function 'main':

pgm2.c:5: error: redefinition of 'y'

pgm2.c:4: note: previous definition of 'y' was here

4. Which of the following is not a valid variable name declaration?

- a) float PI = 3.14;
- b) double PI = 3.14;
- c) int PI = 3.14;
- d) #define PI 3.14

Answer: d

Explanation: #define PI 3.14 is a macro preprocessor, it is a textual substitution.

5. What will happen if the following C code is executed?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int main = 3;
5.      printf("%d", main);
6.      return 0;
7.  }
```

- a) It will cause a compile-time error
- b) It will cause a run-time error
- c) It will run without any error and prints 3
- d) It will experience infinite looping

Answer: c

Explanation: A C program can have same function name and same variable name.

\$ cc pgm3.c

\$ a.out

3

6. What is the problem in the following variable declaration?

float 3Bedroom-Hall-Kitchen?;

- a) The variable name begins with an integer
- b) The special character '-'
- c) The special character '?'
- d) All of the mentioned

Answer: d

Explanation: A variable name cannot start with an integer, along with that the C compiler interprets the '-' and '?' as a minus operator and a question mark operator respectively.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int ThisIsVariableName = 12;
5.      int ThisIsVariablename = 14;
6.      printf("%d", ThisIsVariablename);
7.      return 0;
8.  }

```

- a) The program will print 12
- b) The program will print 14
- c) The program will have a runtime error
- d) The program will cause a compile-time error due to redeclaration

Answer: b

Explanation: Variable names ThisIsVariablename and ThisIsVariableName are both distinct as C is case sensitive.

Output:

\$ cc pgm4.c

\$ a.out

14

8. Which of the following cannot be a variable name in C?

- a) volatile
- b) true
- c) friend
- d) export

Answer: a

Explanation: volatile is C keyword.

Data Types and Sizes – 1

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a[5] = {1, 2, 3, 4, 5};
5.      int i;
6.      for (i = 0; i < 5; i++)
7.          if ((char)a[i] == '5')
8.              printf("%d\n", a[i]);
9.          else
10.             printf("FAIL\n");
11.  }

```

- a) The compiler will flag an error
- b) The program will compile and print the output 5
- c) The program will compile and print the ASCII value of 5
- d) The program will compile and print FAIL for 5 times

Answer: d

Explanation: The ASCII value of 5 is 53, the char type-casted integral value 5 is 5 only.

Output:

\$ cc pgm1.c

\$ a.out

FAIL

FAIL

FAIL

FAIL

FAIL

2. The format identifier '%i' is also used for _____ data type.

a) char

b) int

c) float

d) double

Answer: b

Explanation: Both %d and %i can be used as a format identifier for int data type.

3. Which data type is most suitable for storing a number 65000 in a 32-bit system?

a) signed short

b) unsigned short

c) long

d) int

Answer: b

Explanation: 65000 comes in the range of short (16-bit) which occupies the least memory.

Signed short ranges from -32768 to 32767 and hence we should use unsigned short.

4. Which of the following is a User-defined data type?

a) typedef int Boolean;

b) typedef enum {Mon, Tue, Wed, Thu, Fri} Workdays;

c) struct {char name[10], int age};

d) all of the mentioned

Answer: d

Explanation: typedef and struct are used to define user-defined data types.

5. What is the size of an int data type?

a) 4 Bytes

b) 8 Bytes

c) Depends on the system/compiler

d) Cannot be determined

Answer: c

Explanation: The size of the data types depend on the system.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      signed char chr;
5.      chr = 128;
6.      printf("%d\n", chr);
7.      return 0;
8.  }
```

- a) 128
- b) -128
- c) Depends on the compiler
- d) None of the mentioned

Answer: b

Explanation: signed char will be a negative number.

Output:

\$ cc pgm2.c

\$ a.out

-128

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char c;
5.      int i = 0;
6.      FILE *file;
7.      file = fopen("test.txt", "wt");
8.      fprintf(file, "%c", 'a');
9.      fprintf(file, "%c", -1);
10.     fprintf(file, "%c", 'b');
11.     fclose(file);
12.     file = fopen("test.txt", "r");
13.     while ((c = fgetc(file)) != -1)
14.         printf("%c", c);
15.     return 0;
16. }
```

- a) a
- b) Infinite loop
- c) Depends on what fgetc returns
- d) Depends on the compiler

Answer: a

Explanation: None.

Output:

\$ cc pgm3.c

\$ a.out

a

8. What is short int in C programming?
- a) The basic data type of C
 - b) Qualifier
 - c) Short is the qualifier and int is the basic data type
 - d) All of the mentioned

Answer: c

Explanation: None.

Data Types and Sizes – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      float f1 = 0.1;
5.      if (f1 == 0.1)
6.          printf("equal\n");
7.      else
8.          printf("not equal\n");
9.  }
```

- a) equal
- b) not equal
- c) output depends on the compiler
- d) error

Answer: b

Explanation: 0.1 by default is of type double which has different representation than float resulting in inequality even after conversion.

Output:

\$ cc pgm4.c

\$ a.out

not equal

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      float f1 = 0.1;
5.      if (f1 == 0.1f)
6.          printf("equal\n");
7.      else
8.          printf("not equal\n");
9.  }
```

- a) equal
- b) not equal
- c) output depends on compiler
- d) error

Answer: a

Explanation: 0.1f results in 0.1 to be stored in floating point representations.

Output:

\$ cc pgm5.c

\$ a.out

equal

3. What will be the output of the following C code on a 32-bit machine?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 10000;
5.      double y = 56;
6.      int *p = &x;
7.      double *q = &y;
8.      printf("p and q are %d and %d", sizeof(p), sizeof(q));
9.      return 0;
10. }
```

a) p and q are 4 and 4

b) p and q are 4 and 8

c) compiler error

d) p and q are 2 and 8

Answer: a

Explanation: Size of any type of pointer is 4 on a 32-bit machine.

Output:

\$ cc pgm6.c

\$ a.out

p and q are 4 and 4

4. Which is correct with respect to the size of the data types?

a) char > int > float

b) int > char > float

c) char < int < double

d) double > char > int

Answer: c

Explanation: char has less bytes than int and int has less bytes than double in any system

5. What will be the output of the following C code on a 64 bit machine?

```
1.  #include <stdio.h>
2.  union Sti
3.  {
4.      int nu;
5.      char m;
6.  };
7.  int main()
8.  {
9.      union Sti s;
10.     printf("%d", sizeof(s));
11.     return 0;
12. }
```

- a) 8
- b) 5
- c) 9
- d) 4

Answer: d

Explanation: Since the size of a union is the size of its maximum data type, here int is the largest data type. Hence the size of the union is 4.

Output:

\$ cc pgm7.c

\$ a.out

4

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      float x = 'a';
5.      printf("%f", x);
6.      return 0;
7.  }
```

- a) a
- b) run time error
- c) a.0000000
- d) 97.000000

Answer: d

Explanation: Since the ASCII value of a is 97, the same is assigned to the float variable and printed.

Output:

\$ cc pgm8.c

\$ a.out

97.000000

7. Which of the data types has the size that is variable?

- a) int
- b) struct
- c) float
- d) double

Answer: b

Explanation: Since the size of the structure depends on its fields, it has a variable size.

Constants – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      enum {ORANGE = 5, MANGO, BANANA = 4, PEACH};
5.      printf("PEACH = %d\n", PEACH);
6.  }
```

- a) PEACH = 3
- b) PEACH = 4
- c) PEACH = 5
- d) PEACH = 6

Answer: c

Explanation: In enum, the value of constant is defined to the recent assignment from left.

Output:

\$ cc pgm1.c

\$ a.out

PEACH = 5

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("C programming %s", "Class by\n%s Sanfoundry", "WOW");
5.  }
```

- a)
C programming Class by
WOW Sanfoundry
- b) C programming Class by\n%s Sanfoundry
- c)
C programming Class by
%s Sanfoundry
- d) Compilation error

Answer: c

Explanation: This program has only one %s within first double quotes, so it does not read the string "WOW".

The %s along with the Sanfoundry is not read as a format modifier while new line character prints the new line.

Output:

\$ cc pgm2.c

\$ a.out

C programming Class by

%s Sanfoundry

3. In the following code snippet, character pointer str holds a reference to the string

```
char *str = "Sanfoundry.com\0" "training classes";
```

- a) Sanfoundry.com
- b) Sanfoundry.com\0training classes
- c) Sanfoundry.comtraining classes
- d) Invalid declaration

Answer: b

Explanation: '\0' is accepted as a char in the string. Even though strlen will give length of string "Sanfoundry.com", in memory str is pointing to entire string including training classes.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define a 10
3.  int main()
4.  {
5.      const int a = 5;
6.      printf("a = %d\n", a);
7.  }
```

- a) a = 5
- b) a = 10
- c) Compilation error
- d) Runtime error

Answer: c

Explanation: The #define substitutes a with 10 without leaving any identifier, which results in Compilation error.

Output:

\$ cc pgm3.c

pgm3.c: In function 'main':

pgm3.c:5: error: expected identifier or '(' before numeric constant

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int var = 010;
5.      printf("%d", var);
}
```

- a) 2
- b) 8
- c) 9
- d) 10

Answer: b

Explanation: 010 is octal representation of 8.

Output:

```
$ cc pgm4.c
$ a.out
8
```

6. What will be the output of the following C function?

```
1.  #include <stdio.h>
2.  enum birds {SPARROW, PEACOCK, PARROT};
3.  enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
4.  int main()
5.  {
6.      enum birds m = TIGER;
7.      int k;
8.      k = m;
9.      printf("%d\n", k);
10.     return 0;
11. }
```

- a) 0
- b) Compile time error
- c) 1
- d) 8

Answer: d

Explanation: m is an integer constant, hence it is compatible.

Output:

```
$ cc pgm5.c
$ a.out
8
```

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define MAX 2
3.  enum bird {SPARROW = MAX + 1, PARROT = SPARROW + MAX};
4.  int main()
5.  {
6.      enum bird b = PARROT;
7.      printf("%d\n", b);
8.      return 0;
9.  }
```

- a) Compilation error
- b) 5
- c) Undefined value
- d) 2

Answer: b

Explanation: MAX value is 2 and hence PARROT will have value 3 + 2.

Output:

```
$ cc pgm6.c
$ a.out
5
```

8. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  int main()
4.  {
5.      char *str = "x";
6.      char c = 'x';
7.      char ary[1];
8.      ary[0] = c;
9.      printf("%d %d", strlen(str), strlen(ary));
10.     return 0;
11. }
```

- a) 1 1
- b) 2 1
- c) 2 2
- d) 1 (undefined value)

Answer: d

Explanation: str is null terminated, but ary is not null terminated.

Output:

\$ cc pgm7.c

\$ a.out

1 5

Constants – 2

1. enum types are processed by _____

- a) Compiler
- b) Preprocessor
- c) Linker
- d) Assembler

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("sanfoundry\rclass\n");
5.      return 0;
6.  }
```

- a) sanfoundryclass
- b)
sanfoundry
class

- c) classundry
- d) sanfoundry

View Answer

Answer: c

Explanation: r is carriage return and moves the cursor back. sanfo is replaced by class.

Output:

```
$ cc pgm8.c
```

```
$ a.out
```

```
classundry
```

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      printf("sanfoundry\r\nclass\n");
5.      return 0;
6.  }
```

a) sanfoundryclass

b)

sanfoundry

class

c) classundry

d) sanfoundry

Answer: b

Explanation: rn combination makes the cursor move to the next line.

Output:

```
$ cc pgm9.c
```

```
$ a.out
```

```
sanfoundry
```

```
class
```

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      const int p;
5.      p = 4;
6.      printf("p is %d", p);
7.      return 0;
8.  }
```

a) p is 4

b) Compile time error

c) Run time error

d) p is followed by a garbage value

Answer: b

Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.

Output:

```
$ cc pgm10.c
```

```
pgm10.c: In function 'main':
```

```
pgm10.c:5: error: assignment of read-only variable 'p'
```

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int k = 4;
5.      int *const p = &k;
6.      int r = 3;
7.      p = &r;
8.      printf("%d", p);
9.  }
```

- a) Address of k
- b) Address of r
- c) Compile time error
- d) Address of k + address of r

Answer: c

Explanation: Since the pointer p is declared to be constant, trying to assign it with a new value results in an error.

Output:

```
$ cc pgm11.c
```

```
pgm11.c: In function 'main':
```

```
pgm11.c:7: error: assignment of read-only variable 'p'
```

```
pgm11.c:8: warning: format '%d' expects type 'int', but argument 2 has type 'int * const'
```

6. Which of the following statement is false?

- a) Constant variables need not be defined as they are declared and can be defined later
- b) Global constant variables are initialized to zero
- c) const keyword is used to define constant values
- d) You cannot reassign a value to a constant variable

Answer: a

Explanation: Since the constant variable has to be declared and defined at the same time, not doing it results in an error.

7. What will be the output of the following C code?


```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int const k = 5;
5.      k++;
6.      printf("k is %d", k);
7.  }

```

- a) k is 6
- b) Error due to const succeeding int
- c) Error, because a constant variable can be changed only twice
- d) Error, because a constant variable cannot be changed

Answer: d

Explanation: Constant variable has to be declared and defined at the same time. Trying to change it results in an error.

Output:

\$ cc pgm12.c

pgm12.c: In function 'main':

pgm12.c:5: error: increment of read-only variable 'k'

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int const print()
3.  {
4.      printf("Sanfoundry.com");
5.      return 0;
6.  }
7.  void main()
8.  {
9.      print();
10. }

```

- a) Error because function name cannot be preceded by const
- b) Sanfoundry.com
- c) Sanfoundry.com is printed infinite times
- d) Blank screen; no output

Answer: b

Explanation: None.

Output:

\$ cc pgm13.c

\$ a.out

Sanfoundry.com

Declarations – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void foo(const int *);
3.  int main()
4.  {
5.      const int i = 10;
6.      printf("%d ", i);
7.      foo(&i);
8.      printf("%d", i);
9.
10.     }
11.     void foo(const int *i)
12.     {
13.         *i = 20;
14.     }
```

- a) Compile time error
- b) 10 20
- c) Undefined value
- d) 10

Answer: a

Explanation: Cannot change a const type value.

Output:

\$ cc pgm1.c

pgm1.c: In function 'foo':

pgm1.c:13: error: assignment of read-only location '*i'

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      const int i = 10;
5.      int *ptr = &i;
6.      *ptr = 20;
7.      printf("%d\n", i);
8.      return 0;
9.  }
```

- a) Compile time error
- b) Compile time warning and printf displays 20
- c) Undefined behaviour
- d) 10

Answer: b

Explanation: Changing const variable through non-constant pointers invokes compiler warning.

Output:

\$ cc pgm2.c

pgm2.c: In function 'main':

pgm2.c:5: warning: initialization discards qualifiers from pointer target type

\$ a.out
20

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      j = 10;
5.      printf("%d\n", j++);
6.      return 0;
7.  }
```

- a) 10
- b) 11
- c) Compile time error
- d) 0

Answer: c

Explanation: Variable j is not defined.

Output:

\$ cc pgm3.c

pgm3.c: In function 'main':

pgm3.c:4: error: 'j' undeclared (first use in this function)

pgm3.c:4: error: (Each undeclared identifier is reported only once

pgm3.c:4: error: for each function it appears in.)

4. Will the following C code compile without any error?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      for (int k = 0; k < 10; k++);
5.      return 0;
6.  }
```

- a) Yes
- b) No
- c) Depends on the C standard implemented by compilers
- d) Error

Answer: c

Explanation: Compilers implementing C90 do not allow this, but compilers implementing C99 allow it.

Output:

\$ cc pgm4.c

pgm4.c: In function 'main':

pgm4.c:4: error: 'for' loop initial declarations are only allowed in C99 mode

pgm4.c:4: note: use option -std=c99 or -std=gnu99 to compile your code

5. Will the following C code compile without any error?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int k;
5.      {
6.          int k;
7.          for (k = 0; k < 10; k++);
8.      }
9.  }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the C standard implemented by compilers

Answer: a

Explanation: There can be blocks inside the block. But within a block, variables have only block scope.

Output:

\$ cc pgm5.c

6. Which of the following declaration is not supported by C?

- a) String str;
- b) char *str;
- c) float str = 3e2;
- d) Both String str; & float str = 3e2;

Answer: a

Explanation: It is legal in Java, but not in C.

7. Which of the following format identifier can never be used for the variable var?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *var = "Advanced Training in C by Sanfoundry.com";
5.  }
```

- a) %f
- b) %d
- c) %c
- d) %s

Answer: a

Explanation: %c can be used to print the indexed position.

%d can still be used to display its ASCII value.

%s is recommended.

%f cannot be used for the variable var

Declarations – 2

1. Which of the following declaration is illegal?

- a) `char *str = "Best C programming classes by Sanfoundry";`
- b) `char str[] = "Best C programming classes by Sanfoundry";`
- c) `char str[20] = "Best C programming classes by Sanfoundry";`
- d) `char[] str = "Best C programming classes by Sanfoundry";`

Answer: d

Explanation: `char[] str` is a declaration in Java, but not in C.

2. Which keyword is used to prevent any changes in the variable within a C program?

- a) `immutable`
- b) `mutable`
- c) `const`
- d) `volatile`

Answer: c

Explanation: `const` is a keyword constant in C program.

3. Which of the following is not a pointer declaration?

- a) `char a[10];`
- b) `char a[] = {'1', '2', '3', '4'};`
- c) `char *str;`
- d) `char a;`

Answer: d

Explanation: Array declarations are pointer declarations.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int k = 4;
5.      float k = 4.4;
6.      printf("%d", k)
7.  }
```

- a) Compile time error
- b) 4
- c) 4.000000
- d) 4.4

Answer: a

Explanation: Since the variable `k` is defined both as integer and as float, it results in an error.

Output:

`$ cc pgm8.c`

`pgm8.c: In function 'main':`

`pgm8.c:5: error: conflicting types for 'k'`

pgm8.c:4: note: previous definition of 'k' was here
pgm8.c:6: warning: format '%d' expects type 'int', but argument 2 has type 'double'
pgm8.c:7: error: expected ';' before '}' token

5. Which of the following statement is false?

- a) A variable defined once can be defined again with different scope
- b) A single variable cannot be defined with two different types in the same scope
- c) A variable must be declared and defined at the same time
- d) A variable refers to a location in memory

Answer: c

Explanation: It is not an error if the variable is declared and not defined. For example – extern declarations.

6. A variable declared in a function can be used in main().

- a) True
- b) False
- c) True if it is declared static
- d) None of the mentioned

Answer: b

Explanation: Since the scope of the variable declared within a function is restricted only within that function, so the above statement is false.

7. The name of the variable used in one function cannot be used in another function.

- a) True
- b) False

Answer: b

Explanation: Since the scope of the variable declared within a function is restricted only within that function, the same name can be used to declare another variable in another function.

Arithmetic Operators – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = -3;
5.      int k = i % 2;
6.      printf("%d\n", k);
}
```

- a) Compile time error
- b) -1
- c) 1
- d) Implementation defined

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 3;
5.      int l = i / -2;
6.      int k = i % -2;
7.      printf("%d %d\n", l, k);
8.      return 0;
9.  }
```

- a) Compile time error
- b) -1 1
- c) 1 -1
- d) Implementation defined

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 5;
5.      i = i / 3;
6.      printf("%d\n", i);
7.      return 0;
8.  }
```

- a) Implementation defined
- b) 1
- c) 3
- d) Compile time error

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = -5;
5.      i = i / 3;
6.      printf("%d\n", i);
7.      return 0;
8.  }

```

- a) Implementation defined
- b) -1
- c) -3
- d) Compile time error

Answer: b

Explanation: None.

5. What will be the final value of x in the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 5 * 9 / 3 + 9;
5.  }

```

- a) 3.75
- b) Depends on compiler
- c) 24
- d) 3

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 5.3 % 2;
5.      printf("Value of x is %d", x);
6.  }

```

- a) Value of x is 2.3
- b) Value of x is 1
- c) Value of x is 0.3
- d) Compile time error

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int y = 3;
5.      int x = 5 % 2 * 3 / 2;
6.      printf("Value of x is %d", x);
7.  }
```

- a) Value of x is 1
- b) Value of x is 2
- c) Value of x is 3
- d) Compile time error

Answer: a

Explanation: None.

Arithmetic Operators – 2

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int a = 3;
5.      int b = ++a + a++ + -a;
6.      printf("Value of b is %d", b);
7.  }
```

- a) Value of x is 12
- b) Value of x is 13
- c) Value of x is 10
- d) Undefined behaviour

Answer: d

Explanation: None.

2. What is the precedence of arithmetic operators (from highest to lowest)?

- a) %, *, /, +, -
- b) %, +, /, *, -
- c) +, -, %, *, /
- d) %, +, -, *, /

Answer: a

Explanation: None.

3. Which of the following is not an arithmetic operation?

- a) $a * = 10;$
- b) $a / = 10;$
- c) $a ! = 10;$
- d) $a \% = 10;$

Answer: c

Explanation: None.

4. Which of the following data type will throw an error on modulus operation(%)?

- a) char
- b) short
- c) int
- d) float

Answer: d

Explanation: None.

5. Which among the following are the fundamental arithmetic operators, i.e, performing the desired operation can be done using that operator only?

- a) +, -
- b) +, -, %
- c) +, -, *, /
- d) +, -, *, /, %

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10;
5.      double b = 5.6;
6.      int c;
7.      c = a + b;
8.      printf("%d", c);
}
```

- a) 15
- b) 16
- c) 15.6
- d) 10

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10, b = 5, c = 5;
5.      int d;
6.      d = a == (b + c);
7.      printf("%d", d);
8.  }
```

- a) Syntax error
- b) 1
- c) 10
- d) 5

Answer: b

Explanation: None.

Relational & Logical Operators – 1

1. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 1, y = 0, z = 5;
5.      int a = x && y || z++;
6.      printf("%d", z);
}
```

- a) 6
- b) 5
- c) 0
- d) Varies

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 1, y = 0, z = 5;
5.      int a = x && y && z++;
6.      printf("%d", z);
}
```

- a) 6
- b) 5
- c) 0
- d) Varies

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int x = 1, y = 0, z = 3;
5.      x > y ? printf("%d", z) : return z;
6.  }
```

- a) 3
- b) 1
- c) Compile time error
- d) Run time error

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 1, z = 3;
5.      int y = x << 3;
6.      printf(" %d\n", y);
7.  }
```

- a) -2147483648
- b) -1
- c) Run time error
- d) 8

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      int x = 0, y = 2, z = 3;
5.      int a = x & y | z;
6.      printf("%d", a);
7.  }
```

- a) 3
- b) 0
- c) 2

d) Run time error

Answer: a

Explanation: None.

6. What will be the final value of j in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 0, j = 0;
5.      if (i && (j = i + 10))
6.          //do something
7.          ;
8.  }
```

a) 0

b) 10

c) Depends on the compiler

d) Depends on language standard

Answer: a

Explanation: None.

7. What will be the final value of j in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 10, j = 0;
5.      if (i || (j = i + 10))
6.          //do something
7.          ;
8.  }
```

a) 0

b) 20

c) Compile time error

d) Depends on language standard

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int i = 1;
5.      if (i++ && (i == 1))
6.          printf("Yes\n");
7.      else
8.          printf("No\n");
9.  }

```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

Answer: b

Explanation: None.

Relational & Logical Operators – 2

1. Are logical operator sequence points?

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

Answer: a

Explanation: None.

2. Do logical operators in the C language are evaluated with the short circuit?

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

Answer: a

Explanation: None.

3. What is the result of logical or relational expression in C?

- a) True or False
- b) 0 or 1
- c) 0 if an expression is false and any positive number if an expression is true
- d) None of the mentioned

Answer: b

Explanation: None.

4. What will be the final value of d in the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10, b = 5, c = 5;
5.      int d;
6.      d = b + c == a;
7.      printf("%d", d);
8.  }
```

- a) Syntax error
- b) 1
- c) 5
- d) 10

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10, b = 5, c = 3;
5.      b != !a;
6.      c = !!a;
7.      printf("%d\t%d", b, c);
8.  }
```

1 a) 5

- 3 b) 0
- 3 c) 5
- 1 d) 1

Answer: a

Explanation: None.

6. Which among the following is NOT a logical or relational operator?

- a) !=
- b) ==
- c) ||
- d) =

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a = 10;
5.      if (a == a--)
6.          printf("TRUE 1\t");
7.      a = 10;
8.      if (a == --a)
9.          printf("TRUE 2\t");
10. }

```

- a) TRUE 1
- b) TRUE 2
- TRUE 2 c) TRUE 1
- d) Compiler Dependent

r

Answer: d

Explanation: This is a sequence point problem and hence the result will be implementation dependent.

8. Relational operators cannot be used on _____

- a) structure
- b) long
- c) strings
- d) float

Answer: a

Explanation: None.

Type Conversions – 1

1. What will be the output of the following C code?

```

1.  #include <stdio.h>
2.  void main()
3.  {
4.      float x = 0.1;
5.      if (x == 0.1)
6.          printf("Sanfoundry");
7.      else
8.          printf("Advanced C Classes");
9. }

```

- a) Advanced C Classes
- b) Sanfoundry
- c) Run time error
- d) Compile time error

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      float x = 0.1;
5.      printf("%d, ", x);
6.      printf("%f", x);
7.  }
```

- a) 0.100000, junk value
- b) Junk value, 0.100000
- c) 0, 0.100000
- d) 0, 0.999999

Answer: b

Explanation: None.

3. What will be the output of the following C code? (Initial values: x= 7, y = 8)

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      float x;
5.      int y;
6.      printf("enter two numbers\n", x);
7.      scanf("%f %f", &x, &y);
8.      printf("%f, %d", x, y);
9.  }
```

- a) 7.000000, 7
- b) Run time error
- c) 7.000000, junk
- d) Varies

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  void main()
3.  {
4.      double x = 123828749.66;
5.      int y = x;
6.      printf("%d\n", y);
7.      printf("%lf\n", y);
8.  }
```

- a) 0, 0.0
- b) 123828749, 123828749.66
- c) 12382874, 12382874.0
- d) 123828749, 0.000000

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```
1.      #include <stdio.h>
2.      void main()
3.      {
4.          int x = 97;
5.          char y = x;
6.          printf("%c\n", y);
7.      }
```

- a) a
- b) b
- c) 97
- d) Run time error

Answer: a

Explanation: None.

6. When double is converted to float, then the value is?

- a) Truncated
- b) Rounded
- c) Depends on the compiler
- d) Depends on the standard

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
1.      #include <stdio.h>
2.      int main()
3.      {
4.          unsigned int i = 23;
5.          signed char c = -23;
6.          if (i > c)
7.              printf("Yes\n");
8.          else if (i < c)
9.              printf("No\n");
10.     }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the operating system

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
1.      #include <stdio.h>
2.      int main()
3.      {
4.          int i = 23;
5.          char c = -23;
6.          if (i < c)
7.              printf("Yes\n");
8.          else
9.              printf("No\n");
10.     }
```

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

Answer: b

Explanation: None.

Type Conversions – 2

1. function tolower(c) defined in library <ctype.h> works for _____

- a) Ascii character set
- b) Unicode character set
- c) Ascii and utf-8 but not EBCDIC character set
- d) Any character set

Answer: d

Explanation: None.

2. What will be the output of the following C code considering the size of a short int is 2, char is 1 and int is 4 bytes?

```
1.      #include <stdio.h>
2.      int main()
3.      {
4.          short int i = 20;
5.          char c = 97;
6.          printf("%d, %d, %d\n", sizeof(i), sizeof(c), sizeof(c + i));
7.          return 0;
8.      }
```

- a) 2, 1, 2
- b) 2, 1, 1
- c) 2, 1, 4
- d) 2, 2, 8

Answer: c

Explanation: None.

3. Which type of conversion is NOT accepted?

- a) From char to int
- b) From float to char pointer
- c) From negative int to char
- d) From double to char

Answer: b

Explanation: Conversion of a float to pointer type is not allowed.

4. What will be the data type of the result of the following operation?

```
(float)a * (int)b / (long)c * (double)d
```

- a) int
- b) long
- c) float
- d) double

Answer: d

Explanation: None.

5. Which of the following type-casting have chances for wrap around?

- a) From int to float
- b) From int to char
- c) From char to short
- d) From char to int

Answer: b

Explanation: None.

6. Which of the following typecasting is accepted by C?

- a) Widening conversions
- b) Narrowing conversions
- c) Widening & Narrowing conversions
- d) None of the mentioned

Answer: c

Explanation: None.

7. When do you need to use type-conversions?

- a) The value to be stored is beyond the max limit
- b) The value to be stored is in a form not supported by that data type
- c) To reduce the memory in use, relevant to the value
- d) All of the mentioned

Answer: d

Explanation: None.

Increment and Decrement Operators – 1

1. What is the difference between the following 2 codes?

```
#include <stdio.h> //Program 1
int main()
{
    int d, a = 1, b = 2;
    d = a++ + ++b;
    printf("%d %d %d", d, a, b);
}

#include <stdio.h> //Program 2    int main()
{
    int d, a = 1, b = 2;
    d = a++ +++b;
    printf("%d %d %d", d, a, b);
}
```

- a) No difference as space doesn't make any difference, values of a, b, d are same in both the case
- b) Space does make a difference, values of a, b, d are different
- c) Program 1 has syntax error, program 2 is not
- d) Program 2 has syntax error, program 1 is not

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int a = 1, b = 1, c;
    c = a++ + b;
    printf("%d, %d", a, b);
}
```

- a) a = 1, b = 1
- b) a = 2, b = 1
- c) a = 1, b = 2
- d) a = 2, b = 2

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int a = 1, b = 1, d = 1;
    printf("%d, %d, %d", ++a + ++a+a++, a++ + ++b, ++d + d++ + a++);
}
```

- ```
 }
```
- a) 15, 4, 5
  - b) 9, 6, 9
  - c) 9, 3, 5
  - d) Undefined (Compiler Dependent)

**Answer: d**

Explanation: None.

4. For which of the following, "PI++;" code will fail?

- a) #define PI 3.14
- b) char \*PI = "A";
- c) float PI = 3.14;
- d) none of the Mentioned

**Answer: a**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 10, b = 10;
 if (a = 5)
 b--;
 printf("%d, %d", a, b--);
}
```

- a) a = 10, b = 9
- b) a = 10, b = 8
- c) a = 5, b = 9
- d) a = 5, b = 8

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 int j = i++ + i;
 printf("%d\n", j);
}
```

- a) 0
- b) 1
- c) 2
- d) Compile time error

**Answer: b**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 2;
 int j = ++i + i;
 printf("%d\n", j);
}
```

- a) 6
- b) 5
- c) 4
- d) Compile time error

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 2;
 int i = i++ + i;
 printf("%d\n", i);
}
```

- a) = operator is not a sequence point
- b) ++ operator may return value with or without side effects
- c) it can be evaluated as (i++)+i or i+(++i)
- d) = operator is a sequence point

**Answer: a**

Explanation: None.

## Increment and Decrement Operators – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 int x = i++, y = ++i;
 printf("%d %d\n", x, y);
 return 0;
}
```

- }
  - a) 0, 2
  - b) 0, 1
  - c) 1, 2
  - d) Undefined

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 10;
 int *p = &i;
 printf("%d\n", *p++);
}
```

- a) 10
- b) 11
- c) Garbage value
- d) Address of i

**Answer: a**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 97;
 int y = sizeof(x++);
 printf("X is %d", x);
}
```

- a) X is 97
- b) X is 98
- c) X is 99
- d) Run time error

**Answer: a**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 4, y, z;
 y = --x;
 z = x--;
 printf("%d%d%d", x, y, z);
}
```



- ```

    }
a) 3 2 3
b) 2 3 3
c) 3 2 2
d) 2 3 4

```

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int x = 4;
    int *p = &x;
    int *k = p++;
    int r = p - k;
    printf("%d", r);
}

```

- a) 4
b) 8
c) 1
d) Run time error

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int a = 5, b = -7, c = 0, d;
    d = ++a && ++b || ++c;
    printf("\n%d%d%d%d", a, b, c, d);
}

```

- a) 6 -6 0 0
b) 6 -5 0 1
c) -6 -6 0 1
d) 6 -6 0 1

Answer: d

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int a = -5;
}

```

```

        int k = (a++, ++a);
        printf("%d\n", k);
    }

```

- a) -4
- b) -5
- c) 4
- d) -3

Answer: d

Explanation: None.

Bitwise Operators – 1

1. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int c = 2 ^ 3;
    printf("%d\n", c);
}

```

- a) 1
- b) 8
- c) 9
- d) 0

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    unsigned int a = 10;
    a = ~a;
    printf("%d\n", a);
}

```

- a) -9
- b) -10
- c) -11
- d) 10

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    if (7 & 8)
        printf("Honesty");
}

```

```

        if ((~7 & 0x000f) == 8)
            printf("is the best policy\n");
    }

```

- a) Honesty is the best policy
- b) Honesty
- c) is the best policy
- d) No output

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int a = 2;
    if (a >> 1)
        printf("%d\n", a);
}

```

- a) 0
- b) 1
- c) 2
- d) No Output

Answer: c

Explanation: None.

5. Comment on the output of the following C code.

```

#include <stdio.h>
int main()
{
    int i, n, a = 4;
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        a = a * 2;
}

```

- a) Logical Shift left
- b) No output
- c) Arithmetic Shift right
- d) Bitwise exclusive OR

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 97;
    int y = sizeof(x++);
    printf("x is %d", x);
}
```

- a) x is 97
- b) x is 98
- c) x is 99
- d) Run time error

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 4, y, z;
    y = --x;
    z = x--;
    printf("%d%d%d", x, y, z);
}
```

- a) 3 2 3
- b) 2 2 3
- c) 3 2 2
- d) 2 3 3

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 4;
    int *p = &x;
    int *k = p++;
    int r = p - k;
    printf("%d", r);
}
```

- a) 4
- b) 8
- c) 1
- d) Run time error

Answer: c

Explanation: None.

Bitwise Operators – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int a = 5, b = -7, c = 0, d;
    d = ++a && ++b || ++c;
    printf("\n%d%d%d%d", a, b, c, d);
}
```

- a) 6 -6 0 0
- b) 6 -5 0 1
- c) -6 -6 0 1
- d) 6 -6 0 1

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int a = -5;
    int k = (a++, ++a);
    printf("%d\n", k);
}
```

- a) -3
- b) -5
- c) 4
- d) Undefined

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 2;
    x = x << 1;
    printf("%d\n", x);
}
```

- a) 4
- b) 1
- c) Depends on the compiler

d) Depends on the endianness of the machine

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = -2;
    x = x >> 1;
    printf("%d\n", x);
}
```

- a) 1
- b) -1
- c) $2^{31} - 1$ considering int to be 4 bytes
- d) Either -1 or 1

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    if (~0 == 1)
        printf("yes\n");
    else
        printf("no\n");
}
```

- a) yes
- b) no
- c) compile time error
- d) undefined

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = -2;
    if (!0 == 1)
        printf("yes\n");
    else
        printf("no\n");
}
```

- a) yes
- b) no
- c) run time error
- d) undefined

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int y = 0;
    if (1 |(y = 1))
        printf("y is %d\n", y);
    else
        printf("%d\n", y);
}
```

- a) y is 1
- b) 1
- c) run time error
- d) undefined

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int y = 1;
    if (y & (y = 2))
        printf("true %d\n", y);
    else
        printf("false %d\n", y);
}
```

- a) true 2
- b) false 2
- c) either true 2 or false 2
- d) true 1

Answer: a

Explanation: None.

Assignment Operators & Expressions – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 0;
    if (x = 0)
        printf("Its zero\n");
    else
        printf("Its not zero\n");
}
```

- a) Its not zero
- b) Its zero
- c) Run time error
- d) None

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int k = 8;
    int x = 0 == 1 && k++;
    printf("%d%d\n", x, k);
}
```

- a) 0 9
- b) 0 8
- c) 1 8
- d) 1 9

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    char a = 'a';
    int x = (a % 10)++;
    printf("%d\n", x);
}
```

- a) 6
- b) Junk value
- c) Compile time error
- d) 7

Answer: c

Explanation: None.

4. What will be the output of the following C code snippet?

```
#include <stdio.h>
void main()
{
    1 < 2 ? return 1: return 2;
}
```

- a) returns 1
- b) returns 2
- c) Varies
- d) Compile time error

Answer: d

Explanation: None.

5. What will be the output of the following C code snippet?

```
#include <stdio.h>
void main()
{
    unsigned int x = -5;
    printf("%d", x);
}
```

- a) Run time error
- b) Aries
- c) -5
- d) 5

Answer: c

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 2, y = 1;
    x *= x + y;
    printf("%d\n", x);
    return 0;
}
```

- a) 5
- b) 6
- c) Undefined behaviour
- d) Compile time error

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 2, y = 2;
    x /= x / y;
    printf("%d\n", x);
    return 0;
}
```

- a) 2
- b) 1
- c) 0.5
- d) Undefined behaviour

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 1, y = 0;
    x &&= y;
    printf("%d\n", x);
}
```

- a) Compile time error
- b) 1
- c) 0
- d) Undefined behaviour

Answer: a

Explanation: None.

Assignment Operators & Expressions – 2

1. What is the type of the following assignment expression if x is of type float and y is of type int?

- ```
y = x + y;
```
- a) int
  - b) float
  - c) there is no type for an assignment expression
  - d) double

**Answer: a**

Explanation: None.

2. What will be the value of the following assignment expression?

(x = foo())!= 1 considering foo() returns 2

- a) 2
- b) True
- c) 1
- d) 0

3. Operation "a = a \* b + a" can also be written as \_\_\_\_\_

- a) a \*= b + 1;
- b) (c = a \* b)!= (a = c + a);
- c) a = (b + 1)\* a;
- d) All of the mentioned

**Answer: d**

Explanation: None.

4. What will be the final value of c in the following C statement? (Initial value: c = 2)

1. c <= 1;
- a) c = 1;
  - b) c = 2;
  - c) c = 3;
  - d) c = 4;

**Answer: d**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 1, b = 2;
 a += b -= a;
 printf("%d %d", a, b);
}
```

- a) 1 1
- b) 1 2
- c) 2 1
- d) 2 2

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 4, n, i, result = 0;
 scanf("%d", n);
}
```

```

 for (i = 0; i < n; i++)
 result += a;
 }

```

- a) Addition of a and n
- b) Subtraction of a and n
- c) Multiplication of a and n
- d) Division of a and n

**Answer: c**

Explanation: None.

7. Which of the following is an invalid assignment operator?

- a) a %= 10;
- b) a /= 10;
- c) a |= 10;
- d) None of the mentioned

**Answer: d**

Explanation: None.

## Conditional Expressions – 1

1. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int x = 2, y = 0;
 int z = (y++) ? y == 1 && x : 0;
 printf("%d\n", z);
 return 0;
}

```

- a) 0
- b) 1
- c) Undefined behaviour
- d) Compile time error

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int x = 1;
 int y = x == 1 ? getchar(): 2;
}

```

```
 printf("%d\n", y);
 }
}
```

- a) Compile time error
- b) Whatever character getchar function returns
- c) Ascii value of character getchar function returns
- d) 2

**Answer: c**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 1;
 short int i = 2;
 float f = 3;
 if (sizeof((x == 2) ? f : i) == sizeof(float))
 printf("float\n");
 else if (sizeof((x == 2) ? f : i) == sizeof(short int))
 printf("short int\n");
}
```

- a) float
- b) short int
- c) Undefined behaviour
- d) Compile time error

**Answer: a**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 2;
 int b = 0;
 int y = (b == 0) ? a : (a > b) ? (b = 1) : a;
 printf("%d\n", y);
}
```

- a) Compile time error
- b) 1
- c) 2
- d) Undefined behaviour

**Answer: c**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int y = 1, x = 0;
 int l = (y++, x++) ? y : x;
 printf("%d\n", l);
}
```

- a) 1
- b) 2
- c) Compile time error
- d) Undefined behaviour

**Answer: a**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 8;
 int m = 7;
 int z = k < m ? k++ : m++;
 printf("%d", z);
}
```

- a) 7
- b) 8
- c) Run time error
- d) 15

**Answer: a**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 8;
 int m = 7;
 int z = k < m ? k = m : m++;
 printf("%d", z);
}
```

- a) Run time error
- b) 7
- c) 8
- d) Depends on compiler

**Answer: b**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 1 < 2 ? return 1 : return 2;
}
```

- a) returns 1
- b) returns 2
- c) Varies
- d) Compile time error

**Answer: d**

Explanation: None.

## Conditional Expressions – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 8;
 int m = 7;
 k < m ? k++ : m = k;
 printf("%d", k);
}
```

- a) 7
- b) 8
- c) Compile time error
- d) Run time error

**Answer: c**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 8;
 int m = 7;
 k < m ? k = k + 1 : m = m + 1;
 printf("%d", k);
}
```

- a) Compile time error
- b) 9
- c) 8

d) Run time error

**Answer: a**

Explanation: None.

3. What will be the final values of a and c in the following C statement? (Initial values: a = 2, c = 1)

`c = (c) ? a = 0 : 2;`

a) a = 0, c = 0;

b) a = 2, c = 2;

c) a = 2, c = 2;

d) a = 1, c = 2;

**Answer: a**

Explanation: None.

4. What will be the data type of the following expression? (Initial data type: a = int, var1 = double, var2 = float)

`expression (a < 50)? var1 : var2;`

a) int

b) float

c) double

d) Cannot be determined

**Answer: c**

Explanation: None.

5. Which expression has to be present in the following?

`exp1 ? exp2 : exp3;`

a) exp1

b) exp2

c) exp3

d) all of the mentioned

**Answer: d**

Explanation: None.

6. What will be the final value of c in the following C code snippet? (Initial values: a = 1, b = 2, c = 1)

`c += (-c) ? a : b;`

a) Syntax Error

b) c = 1

c) c = 2

d) c = 3



**Answer: c**

Explanation: None.

7. The following C code can be rewritten as \_\_\_\_\_

`c = (n) ? a : b;`

a)

`if (!n)c = b;  
else c = a;`

b)

`if (n <= 0)c = b;  
else c = a;`

c)

`if (n > 0)c = a;  
else c = b;`

d) All of the mentioned

**Answer: a**

Explanation: None.

## Precedence and Order of Evaluation – 1

1. What will be the output of the following C function?

```
#include <stdio.h>
int main()
{
 reverse(1);
}
void reverse(int i)
{
 if (i > 5)
 exit(0);
 printf("%d\n", i);
 return reverse(i++);
}
```

a) 1 2 3 4 5

b) 1 2 3 4

c) Compile time error

d) Stack overflow

**Answer: d**

Explanation: None.

2. What will be the output of the following C function?

```
#include <stdio.h>
void reverse(int i);
int main()
{
 reverse(1);
}. void reverse(int i). {. if (i > 5)
 return ; printf("%d ", i);
return reverse((i++, i));
}
```

- a) 1 2 3 4 5
- b) Segmentation fault
- c) Compilation error
- d) Undefined behaviour

**Answer: a**

Explanation: None.

3. In expression  $i = g() + f()$ , first function called depends on \_\_\_\_\_

- a) Compiler
- b) Associativity of  $()$  operator
- c) Precedence of  $()$  and  $+$  operator
- d) Left to right of the expression

**Answer: a**

Explanation: None.

4. What will be the final values of  $i$  and  $j$  in the following C code?

```
1. #include <stdio.h>
2. int x = 0;
3. int main()
4. {
5. int i = (f() + g()) || g();
6. int j = g() || (f() + g());
7. }
8. int f()
9. {
10. if (x == 0)
11. return x + 1;
12. else
13. return x - 1;
14. }
15. int g()
16. {
17. return x++;
18. }
```

- a)  $i$  value is 1 and  $j$  value is 1
- b)  $i$  value is 0 and  $j$  value is 0
- c)  $i$  value is 1 and  $j$  value is undefined

d) i and j value are undefined

**Answer: d**

Explanation: None.

5. What will be the final values of i and j in the following C code?

```
1. #include <stdio.h>
2. int x = 0;
3. int main()
4. {
5. int i = (f() + g()) | g(); //bitwise or
6. int j = g() | (f() + g()); //bitwise or
7. }
8. int f()
9. {
10. if (x == 0)
11. return x + 1;
12. else
13. return x - 1;
14. }
15. int g()
16. {
17. return x++;
18. }
```

a) i value is 1 and j value is 1

b) i value is 0 and j value is 0

c) i value is 1 and j value is undefined

d) i and j value are undefined

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4. int x = 2, y = 0;
5. int z = y && (y != 10);
6. printf("%d\n", z);
7. return 0;
8. }
```

a) 1

b) 0

c) Undefined behaviour due to order of evaluation

d) 2

**Answer: b**

Explanation: None.

7. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4. int x = 2, y = 0;
5. int z = (y++) ? 2 : y == 1 && x;
6. printf("%d\n", z);
7. return 0;
8. }
```

- a) 0
- b) 1
- c) 2
- d) Undefined behaviour

**Answer: b**

Explanation: None.

8. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4. int x = 2, y = 0;
5. int z;
6. z = (y++, y);
7. printf("%d\n", z);
8. return 0;
9. }
```

- a) 0
- b) 1
- c) Undefined behaviour
- d) Compilation error

**Answer: b**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 2, y = 0, l;
 int z;
 z = y = 1, l = x && y;
 printf("%d\n", l);
 return 0;
}
```

- a) 0
- b) 1

- c) Undefined behaviour due to order of evaluation can be different
- d) Compilation error

**Answer: b**

Explanation: None.

10. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int y = 2;
 int z = y +(y = 10);
 printf("%d\n", z);
}
```

- a) 12
- b) 20
- c) 4
- d) Either 12 or 20

**Answer: b**

Explanation: None.

## Precedence and Order of Evaluation – 4

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int b = 5 - 4 + 2 * 5;
 printf("%d", b);
}
```

- a) 25
- b) -5
- c) 11
- d) 16

**Answer: c**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int b = 5 & 4 & 6;
 printf("%d", b);
}
```

- a) 5
- b) 6
- c) 3
- d) 4

**Answer: d**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int b = 5 & 4 | 6;
 printf("%d", b);
}
```

- a) 6
- b) 4
- c) 1
- d) 0

**Answer: a**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int b = 5 + 7 * 4 - 9 * (3, 2);
 printf("%d", b);
}
```

- a) 6
- b) 15
- c) 13
- d) 21

**Answer: b**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int h = 8;
 int b = (h++, h++);
 printf("%d%d\n", b, h);
}
```

- a) 10 10
- b) 10 9
- c) 9 10

d) 8 10

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int h = 8; int b = h++ + h++ + h++;
 printf("%d\n", h);
}
```

- a) 9
- b) 10
- c) 12
- d) 11

**Answer: d**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int h = 8;. int b = 4 * 6 + 3 * 4 < 3 ? 4 : 3;
 printf("%d\n", b);
}
```

- a) 3
- b) 33
- c) 34
- d) Run time error

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int a = 2 + 3 - 4 + 8 - 5 % 4;
 printf("%d\n", a);
}
```

- a) 0
- b) 8
- c) 11
- d) 9

**Answer: b**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 char a = '0';
 char b = 'm';
 int c = a && b || '1';
 printf("%d\n", c);
}
```

- a) 0
- b) a
- c) 1
- d) m

**Answer: c**

Explanation: None.

10. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 char a = 'A';
 char b = 'B';
 int c = a + b % 3 - 3 * 2;
 printf("%d\n", c);
}
```

- a) 65
- b) 58
- c) 64
- d) 59

**Answer: d**

Explanation: None.

## Precedence and Order of Evaluation – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 2, y = 2;
 float f = y + x /= x / y;
 printf("%d %f\n", x, f);
}
```



```
 return 0;
 }
a) 2 4.000000
b) Compile time error
c) 2 3.500000
d) Undefined behaviour
```

**Answer: b**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 1, y = 2;
 if (x && y == 1)
 printf("true\n");
 else
 printf("false\n");
}
```

- a) true
- b) false
- c) compile time error
- d) undefined behaviour

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 1, y = 2;
 int z = x & y == 2;
 printf("%d\n", z);
}
```

- a) 0
- b) 1
- c) Compile time error
- d) Undefined behaviour

**Answer: b**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 3, y = 2;
 int z = x /= y %= 2;
 printf("%d\n", z);
}
```

- a) 1
- b) Compile time error
- c) Floating point exception
- d) Segmentation fault

**Answer: c**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 3, y = 2;
 int z = x << 1 > 5;
 printf("%d\n", z);
}
```

- a) 1
- b) 0
- c) 3
- d) Compile time error

**Answer: a**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 3; //, y = 2;
 const int *p = &x;
 *p++;
 printf("%d\n", *p);
}
```

- a) Increment of read-only location compile error
- b) 4
- c) Some garbage value
- d) Undefined behaviour

**Answer: c**

Explanation: None.

7. What will be the output of the following C code?

```
1. #include <stdio.h>
2. int main()
3. {
4. int x = 2, y = 2;
5. int z = x ^ y & 1;
6. printf("%d\n", z);
7. }
```

- a) 1
- b) 2
- c) 0
- d) 1 or 2

**Answer: b**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 2, y = 0;
 int z = x && y = 1;
 printf("%d\n", z);
}
```

- a) 0
- b) 1
- c) Compile time error
- d) 2

**Answer: c**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 0, y = 2;
 if (!x && y)
 printf("true\n");
 else
 printf("false\n");
}
```

- a) True
- b) False
- c) Compile time error
- d) Undefined behaviour

**Answer: a**

Explanation: None.

10. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 0, y = 2;
 int z = ~x & y;
 printf("%d\n", z);
}
```

- a) -1
- b) 2
- c) 0
- d) Compile time error

**Answer: b**

Explanation: None

## Precedence and Order of Evaluation – 5

1. Which of the following operators has an associativity from Right to Left?

- a) <=
- b) <<
- c) ==
- d) +=

**Answer: d**

Explanation: None.

2. Which operators of the following have same precedence?

P. "!=", Q. "+=", R. "<=<="

- a) P and Q
- b) Q and R
- c) P and R
- d) P, Q and R

**Answer: b**

Explanation: None.

3. Comment on the following statement.

```
n = 1;
printf("%d, %dn", 3*n, n++);
```

- a) Output will be 3, 2
- b) Output will be 3, 1
- c) Output will be 6, 1
- d) Output is compiler dependent

**Answer: d**

Explanation: None.

4. Which of the following option is the correct representation of the following C statement?

$e = a * b + c / d * f;$

a)  $e = (a * (b + (c / (d * f))));$

b)  $e = ((a * b) + (c / (d * f)));$

c)  $e = ((a * b) + ((c / d) * f));$

d) Both  $e = ((a * b) + (c / (d * f)));$  and  $e = ((a * b) + ((c / d) * f));$

**Answer: d**

Explanation: Verified by  $e = 1 * 2 + 3 / 4 * 5;$  and then using respective braces according to the option.

5. While swapping 2 numbers what precautions to be taken care?

$b = (b / a);$

$a = a * b;$

$b = a / b;$

a) Data type should be either of short, int and long

b) Data type should be either of float and double

c) All data types are accepted except for (char \*)

**d) This code doesn't swap 2 numbers**

**Answer: b**

Explanation: None.

6. What will be the output of the following C code?

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
 int a = 1, b = 2, c = 3, d = 4, e;
```

```
 e = c + d = b * a;
```

```
 printf("%d, %d\n", e, d);
```

```
}
```

a) 7, 4

b) 7, 2

c) 5, 2

d) Syntax error.

**Answer: d**

Explanation: None.

7. Which of the following is the correct order of evaluation for the given expression?

$a = w \% x / y * z;$

a)  $\% / * =$

b)  $/ * \% =$

c)  $= \% * /$

d)  $* \% / =$

**Answer: a**

Explanation: None.

8. Which function in the following expression will be called first?

a = func3(6) - func2(4, 5) / func1(1, 2, 3);

- a) func1();
- b) func2();
- c) func3();
- d) Cannot be predicted

**Answer: d**

Explanation: None.

9. Which of the following operator has the highest precedence in the following?

- a) ()
- b) sizeof
- c) \*
- d) +

**Answer: a**

Explanation: None.

10. Which of the following is a ternary operator?

- a) &&
- b) >>=
- c) ?:
- d) ->

**Answer: c**

Explanation: None.

## Precedence and Order of Evaluation – 3

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int a = 5 * 3 + 2 - 4;
 printf("%d", a);
}
```

- a) 13
- b) 14
- c) 12
- d) 1 6

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int a = 2 + 4 + 3 * 5 / 3 - 5;
 printf("%d", a);
}
```

- a) 7
- b) 6
- c) 10
- d) 9

**Answer: b**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int a = 5 * 3 % 6 - 8 + 3;
 printf("%d", a);
}
```

- a) 10
- b) 2
- c) -2
- d) -3

**Answer: c**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int b = 6;
 int c = 7;
 int a = ++b + c--;
 printf("%d", a);
}
```

- a) Run time error
- b) 15
- c) 13
- d) 14

**Answer: d**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void main(
{
 double b = 8;
 b++;
 printf("%lf", b);
}
```

- a) 9.000000
- b) 9
- c) 9.0
- d) Run time error

**Answer: a**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double b = 3 % 0 * 1 - 4 / 2;
 printf("%lf", b);
}
```

- a) -2
- b) Floating point Exception
- c) 1
- d) 0

**Answer: b**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double b = 5 % 3 & 4 + 5 * 6;
 printf("%lf", b);
}
```

- a) 2
- b) 30
- c) 2.000000
- d) Run time error

**Answer: c**

Explanation: None.



8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double b = 3 && 5 & 4 % 3;
 printf("%lf", b);
}
```

- a) 3.000000
- b) 4.000000
- c) 5.000000
- d) 1.000000

**Answer: d**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double b = 5 & 3 && 4 || 5 | 6;
 printf("%lf", b);
}
```

- a) 1.000000
- b) 0.000000
- c) 7.000000
- d) 2.000000

**Answer: a**

Explanation: None.

10. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 0;
 double b = k++ + ++k + k--;
 printf("%d", k);
}
```

- a) 6
- b) 1
- c) 5
- d) undefined

**Answer: d**

Explanation: None.

## Precedence and Order of Evaluation – 6

1. Which of the following are unary operators?

- a) sizeof
- b) –
- c) ++
- d) all of the mentioned

**Answer: d**

Explanation: None.

2. Where in C the order of precedence of operators do not exist?

- a) Within conditional statements, if, else
- b) Within while, do-while
- c) Within a macro definition
- d) None of the mentioned

**Answer: d**

Explanation: None.

3. Associativity of an operator is \_\_\_\_\_

- a) Right to Left
- b) Left to Right
- c) Random fashion
- d) Both Right to Left and Left to Right

**Answer: d**

Explanation: None.

4. Which of the following method is accepted for assignment?

- a)  $5 = a = b = c = d;$
- b)  $a = b = c = d = 5;$
- c)  $a = b = 5 = c = d;$
- d) None of the mentioned

**Answer: b**

Explanation: None.

5. Which of the following is NOT possible with any 2 operators in C?

- a) Different precedence, same associativity
- b) Different precedence, different associativity
- c) Same precedence, different associativity
- d) All of the mentioned

**Answer: c**

Explanation: None.

6. Which of the following is possible with any 2 operators in C?

- a) Same associativity, different precedence
- b) Same associativity, same precedence
- c) Different associativity, different precedence

d) All of the mentioned

**Answer: d**

Explanation: None.

7. Which of the following operators has the lowest precedence?

- a) !=
- b) &&
- c) ?:
- d) ,

**Answer: d**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 3, i = 0;
 do {
 x = x++;
 i++;
 } while (i != 3);
 printf("%d\n", x);
}
```

- a) Undefined behaviour
- b) Output will be 3
- c) Output will be 6
- d) Output will be 5

**Answer: c**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = -1, b = 4, c = 1, d;
 d = ++a && ++b || ++c;
 printf("%d, %d, %d, %d\n", a, b, c, d);
 return 0;
}
```

- a) 0, 4, 2, 1
- b) 0, 5, 2, 1
- c) -1, 4, 1, 1
- d) 0, 5, 1, 0

**Answer: a**

Explanation: None.

10. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int p = 10, q = 20, r;
 if (r = p = 5 || q > 20)
 printf("%d", r);
 else
 printf("No Output\n");
}
```

- a) 1
- b) 10
- c) 20
- d) No Output

**Answer: a**

Explanation: None.

## If-then-else Statements – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 5;
 if (x < 1)
 printf("hello");
 if (x == 5)
 printf("hi");
 else
 printf("no");
}
```

- a) hi
- b) hello
- c) no
- d) error

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int x;
void main()
{
 if (x)
 printf("hi");
 else
 printf("how are u");
}
```

- a) hi
- b) how are you
- c) compile time error
- d) error

**Answer: b**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 5;
 if (true);
 printf("hello");
}
```

- a) It will display hello
- b) It will throw an error
- c) Nothing will be displayed
- d) Compiler dependent

**Answer: b**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 0;
 if (x == 0)
 printf("hi");
 else
 printf("how are u");
 printf("hello");
}
```

- a) hi
- b) how are you
- c) hello
- d) hihello

**Answer: d**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int x = 5;
```

```
if (x < 1);
 printf("Hello");
```

```
}
```

- a) Nothing
- b) Run time error
- c) Hello
- d) Varies

**Answer: c**

Explanation: None.

6. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
void main()
{
 double ch;
 printf("enter a value between 1 to 2:");
 scanf("%lf", &ch);
 switch (ch)
 {
 case 1:
 printf("1");
 break;
 case 2:
 printf("2");
 break;
 }
}
```

- a) Compile time error
- b) 1
- c) 2
- d) Varies

**Answer: a**

Explanation: None.

7. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
void main()
{
 char *ch;
 printf("enter a value between 1 to 3:");
 scanf("%s", ch);
 switch (ch)
 {
```

```

 case "1":
 printf("1");
 break;
 case "2":
 printf("2");
 break;
 }
}

```

a) 1  
 b) 2  
 c) Compile time error  
 d) No Compile time error

**Answer: c**

Explanation: None.

8. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

#include <stdio.h>
void main()
{
 int ch;
 printf("enter a value between 1 to 2:");
 scanf("%d", &ch);
 switch (ch)
 {
 case 1:
 printf("1\n");
 default:
 printf("2\n");
 }
}

```

a) 1  
 b) 2  
 c) 1 2  
 d) Run time error

**Answer: c**

Explanation: None.

9. What will be the output of the following C code? (Assuming that we have entered the value 2 in the standard input)

```

#include <stdio.h>
void main()
{
 int ch;
 printf("enter a value between 1 to 2:");
 scanf("%d", &ch);
 switch (ch)

```

```

 {
 case 1:
 printf("1\n");
 break;
 printf("Hi");
 default:
 printf("2\n");
 }
}

```

a) 1  
 b) Hi 2  
 c) Run time error  
 d) 2

**Answer: d**

Explanation: None.

10. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

#include <stdio.h>
void main()
{
 int ch;
 printf("enter a value between 1 to 2:");
 scanf("%d", &ch);
 switch (ch, ch + 1)
 {
 case 1:
 printf("1\n");
 break;
 case 2:
 printf("2");
 break;
 }
}

```

a) 1  
 b) 2  
 c) 3  
 d) Run time error

**Answer: b**

Explanation: None.



## If-then-else Statements – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 1;
 if (x > 0)
 printf("inside if\n");
 else if (x > 0)
 printf("inside elseif\n");
}
```

- a) inside if
- b) inside elseif
- c) inside if  
inside elseif
- d) compile time error

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 0;
 if (x++)
 printf("true\n");
 else if (x == 1)
 printf("false\n");
}
```

- a) true
- b) false
- c) compile time error
- d) undefined behaviour

**Answer: b**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int x = 0;
 if (x == 1)
 if (x == 0)
 printf("inside if\n");
 else
```

```

 printf("inside else if\n");
 else
 printf("inside else\n");
}

```

- a) inside if
- b) inside else if
- c) inside else
- d) compile time error

Answer: c

Explanation: None.

4. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int x = 0;
 if (x == 0)
 printf("true, ");
 else if (x = 10)
 printf("false, ");
 printf("%d\n", x);
}

```

- a) false, 0
- b) true, 0
- c) true, 10
- d) compile time error

**Answer: b**

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int x = 0;
 if (x == 1)
 if (x >= 0)
 printf("true\n");
 else
 printf("false\n");
}

```

- a) true
- b) false
- c) Depends on the compiler
- d) No print statement

**Answer: d**

Explanation: None.

6. The C statement "if (a == 1 || b == 2) {}" can be re-written as \_\_\_\_\_

- a)  
if (a == 1)  
if (b == 2){}
- b)  
if (a == 1){}  
if (b == 2){}
- c)  
if (a == 1){}  
else if (b == 2){}
- d) none of the mentioned

**Answer: d**

Explanation: None.

7. Which of the following is an invalid if-else statement?

- a) if (if (a == 1)){}
- b) if (func1 (a)){}
- c) if (a){}
- d) if ((char) a){}

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 1;
 if (a--)
 printf("True");
 if (a++)
 printf("False");
}
```

- a) True
- b) False
- c) True False
- d) No Output

**Answer: a**

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 1;
 if (a)
 printf("All is Well ");
 printf("I am Well\n");
}
```

- ```

        else
            printf("I am not a River\n");
    }

```
- a) Output will be All is Well I am Well
 - b) Output will be I am Well I am not a River
 - c) Output will be I am Well
 - d) Compile time errors during compilation

Answer: d

Explanation: None.

10. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    if (printf("%d", printf(")))
        printf("We are Happy");
    else if (printf("1"))
        printf("We are Sad");
}

```

- a) 0We are Happy
- b) 1We are Happy
- c) 1We are Sad
- d) compile time error

Answer: d

Explanation: None.

Switch Statements – 1

1. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

#include <stdio.h>
void main()
{
    double ch;
    printf("enter a value between 1 to 2:");
    scanf("%lf", &ch);
    switch (ch)
    {
        case 1:
            printf("1");
            break;
        case 2:
            printf("2");
            break;
    }
}

```

- a) Compile time error
- b) 1
- c) 2
- d) Varies

Answer: a

Explanation: None.

2. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
void main()
{
    char *ch;
    printf("enter a value between 1 to 3:");
    scanf("%s", ch);
    switch (ch)
    {
        case "1":
            printf("1");
            break;
        case "2":
            printf("2");
            break;
    }
}
```

- a) 1
- b) Compile time error
- c) 2
- d) Run time error

View Answer

Answer: b

Explanation: None.

3. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
void main()
{
    int ch;
    printf("enter a value between 1 to 2:");
    scanf("%d", &ch);
    switch (ch)
    {
        case 1:
            printf("1\n");
        default:
            printf("2\n");
    }
}
```

- ```

 }
}
a) 1
b) 2
c) 1 2
d) Run time error

```

**Answer: c**

Explanation: None.

4. What will be the output of the following C code? (Assuming that we have entered the value 2 in the standard input)

```

#include <stdio.h>
void main()
{
 int ch;
 printf("enter a value between 1 to 2:");
 scanf("%d", &ch);
 switch (ch)
 {
 case 1:
 printf("1\n");
 break;
 printf("hi");
 default:
 printf("2\n");
 }
}

```

- a) 1  
b) hi 2  
c) Run time error  
d) 2

**Answer: d**

Explanation: None.

5. What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```

#include <stdio.h>
void main()
{
 int ch;
 printf("enter a value between 1 to 2:");
 scanf("%d", &ch);
 switch (ch, ch + 1)
 {
 case 1:
 printf("1\n");

```

```

 break;
 case 2:
 printf("2");
 break;
 }
}

```

a) 1  
 b) 2  
 c) 3  
 d) Run time error

**Answer: b**

Explanation: None.

6. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int a = 1, b = 1;
 switch (a)
 {
 case a*b:
 printf("yes ");
 case a-b:
 printf("no\n");
 break;
 }
}

```

a) yes  
 b) no  
 c) Compile time error  
 d) yes no

**Answer: c**

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int x = 97;
 switch (x)
 {
 case 'a':
 printf("yes ");
 break;
 case 97:
 printf("no\n");
 break;
 }
}

```

- ```
}
a) yes
b) yes no
c) Duplicate case value error
d) Character case value error
```

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    float f = 1;
    switch (f)
    {
        case 1.0:
            printf("yes\n");
            break;
        default:
            printf("default\n");
    }
}
```

- a) yes
b) yes default
c) Undefined behaviour
d) Compile time error

Answer: d

Explanation: None.

Switch Statements – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
const int a = 1, b = 2;
int main()
{
    int x = 1;
    switch (x)
    {
        case a:
            printf("yes ");
        case b:
            printf("no\n");
            break;
    }
}
```


- a) yes no
- b) yes
- c) no
- d) Compile time error

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
#define max(a) a
int main()
{
    int x = 1;
    switch (x)
    {
        case max(2):
            printf("yes\n");
        case max(1):
            printf("no\n");
            break;
    }
}
```

- a) yes no
- b) yes
- c) no
- d) Compile time error

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    switch (printf("Do"))
    {
        case 1:
            printf("First\n");
            break;
        case 2:
            printf("Second\n");
            break;
        default:
            printf("Default\n");
            break;
    }
}
```

- a) Do
- b) DoFirst
- c) DoSecond
- d) DoDefault

Answer: c

Explanation: None.

4. Comment on the output of the following C code.

```
#include <stdio.h>
int main()
{
    int a = 1;
    switch (a)
    case 1:
        printf("%d", a);
    case 2:
        printf("%d", a);
    case 3:
        printf("%d", a);
    default:
        printf("%d", a);
}
```

- a) No error, output is 1111
- b) No error, output is 1
- c) Compile time error, no break statements
- d) Compile time error, case label outside switch statement

Answer: d

Explanation: None.

5. Which datatype can accept the switch statement?

- a) int
- b) char
- c) long
- d) all of the mentioned

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int a = 1;
    switch (a)
    {
        case a:
            printf("Case A ");
        default:
            printf("Default");
    }
```

- ```

 }
}

```
- a) Output: Case A
  - b) Output: Default
  - c) Output: Case A Default
  - d) Compile time error

**Answer: d**

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
switch (ch)
{
 case 'a':
 case 'A':
 printf("true");
}

```

- a) if (ch == 'a' && ch == 'A') printf("true");
- b) if (ch == 'a') printf("true");
- c) if (ch == 'a' || ch == 'A') printf("true");
- d) none of the mentioned

**Answer: c**

Explanation: None.

## For Loops – 1

1. The C code 'for(;;)' represents an infinite loop. It can be terminated by \_\_\_\_\_

- a) break
- b) exit(0)
- c) abort()
- d) terminate

**Answer: a**

Explanation: None.

2. What will be the correct syntax for running two variable for loop simultaneously?

- a)
 

```

for (i = 0; i < n; i++)
for (j = 0; j < n; j += 5)

```
- b)
 

```

for (i = 0, j = 0; i < n, j < n; i++, j += 5)

```
- c)
 

```

for (i = 0; i < n; i++) {}
for (j = 0; j < n; j += 5) {}

```

d) none of the mentioned

**Answer: b**

Explanation: None.

3. Which for loop has range of similar indexes of 'i' used in for (i = 0; i < n; i++)?

a) for (i = n; i > 0; i--)

b) for (i = n; i >= 0; i--)

c) for (i = n-1; i > 0; i--)

d) for (i = n-1; i > -1; i--)

**Answer: d**

Explanation: None.

4. Which of the following cannot be used as LHS of the expression in for (exp1; exp2; exp3)?

a) variable

b) function

c) typedef

d) macros

**Answer: d**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 short i;
 for (i = 1; i >= 0; i++)
 printf("%d\n", i);
}
```

a) The control won't fall into the for loop

b) Numbers will be displayed until the signed limit of short and throw a runtime error

c) Numbers will be displayed until the signed limit of short and program will successfully terminate

d) This program will get into an infinite loop and keep printing numbers with no errors

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 0;
 for (k)
 printf("Hello");
}
```

- a) Compile time error
- b) hello
- c) Nothing
- d) Varies

**Answer: a**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = 0;
 for (k < 3; k++)
 printf("Hello");
}
```

- a) Compile time error
- b) Hello is printed thrice
- c) Nothing
- d) Varies

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double k = 0;
 for (k = 0.0; k < 3.0; k++)
 printf("Hello");
}
```

- a) Run time error
- b) Hello is printed thrice
- c) Hello is printed twice
- d) Hello is printed infinitely

**Answer: b**

Explanation: None.

## For Loops – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 double k = 0;
 for (k = 0.0; k < 3.0; k++);
 printf("%lf", k);
}
```

- a) 2.000000
- b) 4.000000
- c) 3.000000
- d) Run time error

**Answer: c**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k;
 for (k = -3; k < -5; k++)
 printf("Hello");
}
```

- a) Hello
- b) Infinite hello
- c) Run time error
- d) Nothing

**Answer: d**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 for (; ;)
 printf("In for loop\n");
 printf("After loop\n");
}
```

- a) Compile time error
- b) Infinite loop
- c) After loop
- d) Undefined behaviour

**Answer: a**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 for (i++; i == 1; i = 2)
 printf("In for loop ");
}
```

```

 printf("After loop\n");
 }
a) In for loop after loop
b) After loop
c) Compile time error
d) Undefined behaviour

```

**Answer: a**

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int i = 0;
 for (foo(); i == 1; i = 2)
 printf("In for loop\n");
 printf("After loop\n");
}
int foo()
{
 return 1;
}

```

- a) After loop
- b) In for loop after loop
- c) Compile time error
- d) Infinite loop

**Answer: a**

Explanation: None.

6. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int *p = NULL;
 for (foo(); p; p = 0)
 printf("In for loop\n");
 printf("After loop\n");
}

```

- a) In for loop after loop
- b) Compile time error
- c) Infinite loop
- d) Depends on the value of NULL

**Answer: b**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 for (int i = 0; i < 1; i++)
 printf("In for loop\n");
}
```

- a) Compile time error
- b) In for loop
- c) Depends on the standard compiler implements
- d) Depends on the compiler

**Answer: c**

Explanation: None.

## While Loops – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 while ()
 printf("In while loop ");
 printf("After loop\n");
}
```

- a) In while loop after loop
- b) After loop
- c) Compile time error
- d) Infinite loop

**Answer: c**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 do
 printf("In while loop ");
 while (0);
 printf("After loop\n");
}
```

- a) In while loop
- b) In while loop  
after loop



- c) After loop
- d) Infinite loop

**Answer: b**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 do {
 i++;
 printf("In while loop\n");
 } while (i < 3);
}
```

- a)
  - In while loop
  - In while loop
  - In while loop
- b)
  - In while loop
  - In while loop
- c) Depends on the compiler
- d) Compile time error

**Answer: a**

Explanation: None.

4. How many times i value is checked in the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 do {
 i++;
 printf("in while loop\n");
 } while (i < 3);
}
```

- a) 2
- b) 3
- c) 4
- d) 1

**Answer: b**

Explanation: None.

5. How many times i value is checked in the following C code?

```
#include <stdio.h>
int main()
{
 int i = 0;
 while (i < 3)
 i++;
 printf("In while loop\n");
}
```

- a) 2
- b) 3
- c) 4
- d) 1

**Answer: c**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 2;
 do
 {
 printf("Hi");
 } while (i < 2)
}
```

- a) Compile time error
- b) Hi Hi
- c) Hi
- d) Varies

**Answer: a**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 0;
 while (++i)
 {
 printf("H");
 }
}
```

- a) H
- b) H is printed infinite times
- c) Compile time error
- d) Varies

**Answer: b**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 0;
 do
 {
 printf("Hello");
 } while (i != 0);
}
```

- a) Nothing
- b) H is printed infinite times
- c) Hello
- d) Run time error

**Answer: c**

Explanation: None.

## While Loops – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 char *str = "";
 do
 {
 printf("hello");
 } while (str);
}
```

- a) Nothing
- b) Run time error
- c) Varies
- d) Hello is printed infinite times

**Answer: d**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 0;
 while (i < 10)
 {
 i++;
 }
}
```

```

printf("hi\n");
while (i < 8)
{
 i++;
 printf("hello\n");
}
}

```

- a) Hi is printed 8 times, hello 7 times and then hi 2 times
- b) Hi is printed 10 times, hello 7 times
- c) Hi is printed once, hello 7 times
- d) Hi is printed once, hello 7 times and then hi 2 times

**Answer: d**

Explanation: None.

3. What is an example of iteration in C?

- a) for
- b) while
- c) do-while
- d) all of the mentioned

**Answer: d**

Explanation: None.

4. How many times while loop condition is tested in the following C code snippets, if i is initialized to 0 in both the cases?

```

while (i < n)
 i++;

do
 i++;
while (i <= n);

```

- a) n, n
- b) n, n+1
- c) n+1, n
- d) n+1, n+1

**Answer: d**

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int i = 0;
 while (i = 0)

```

- ```
        printf("True\n");
        printf("False\n");
    }
a) True (infinite time)
b) True (1 time) False
c) False
d) Compiler dependent
```

Answer: c

Explanation: None.

6. What will be the output of the following C code?

- ```
#include <stdio.h>
int main()
{
 int i = 0, j = 0;
 while (i < 5, j < 10)
 {
 i++;
 j++;
 }
 printf("%d, %d\n", i, j);
}
a) 5, 5
b) 5, 10
c) 10, 10
d) Syntax error
```

**Answer: c**

Explanation: None.

7. Which loop is most suitable to first perform the operation and then test the condition?

- a) for loop
- b) while loop
- c) do-while loop
- d) none of the mentioned

**Answer: c**

Explanation: None.

## Break and Continue – 1

1. Which keyword can be used for coming out of recursion?

- a) break
- b) return
- c) exit

d) both break and return

**Answer: b**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 0, i = 0, b;
 for (i = 0; i < 5; i++)
 {
 a++;
 continue;
 }
}
```

- a) 2
- b) 3
- c) 4
- d) 5

**Answer: d**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 int a = 0, i = 0, b;
 for (i = 0; i < 5; i++)
 {
 a++;
 if (i == 3)
 break;
 }
}
```

- a) 1
- b) 2
- c) 3
- d) 4

**Answer: d**

Explanation: None.

4. The keyword 'break' cannot be simply used within \_\_\_\_\_

- a) do-while
- b) if-else
- c) for
- d) while

**Answer: b**

Explanation: None.

5. Which keyword is used to come out of a loop only for that iteration?

- a) break
- b) continue
- c) return
- d) none of the mentioned

**Answer: b**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 0, j = 0;
 for (i = 0; i < 5; i++)
 {
 for (j = 0; j < 4; j++)
 {
 if (i > 1)
 break;
 }
 printf("Hi \n");
 }
}
```

- a) Hi is printed 5 times
- b) Hi is printed 9 times
- c) Hi is printed 7 times
- d) Hi is printed 4 times

**Answer: a**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int i = 0;
 int j = 0;
 for (i = 0; i < 5; i++)
 {
 for (j = 0; j < 4; j++)
 {
 if (i > 1)
 continue;
 printf("Hi \n");
 }
 }
}
```

- ```

    }
}
}

```
- a) Hi is printed 9 times
 - b) Hi is printed 8 times
 - c) Hi is printed 7 times
 - d) Hi is printed 6 times

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int i = 0;
    for (i = 0; i < 5; i++)
        if (i < 4)
        {
            printf("Hello");
            break;
        }
}

```

- a) Hello is printed 5 times
- b) Hello is printed 4 times
- c) Hello
- d) Hello is printed 3 times

Answer: c

Explanation: None.

Break and Continue – 2

1. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int i = 0;
    if (i == 0)
    {
        printf("Hello");
        continue;
    }
}

```

- a) Hello is printed infinite times
- b) Hello
- c) Varies
- d) Compile time error

Answer: d

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int i = 0;
    if (i == 0)
    {
        printf("Hello");
        break;
    }
}
```

- a) Hello is printed infinite times
- b) Hello
- c) Varies
- d) Compile time error

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int i = 0;
    do
    {
        i++;
        if (i == 2)
            continue;
        printf("In while loop ");
    } while (i < 2);
    printf("%d\n", i);
}
```

- a) In while loop 2
- b) In while loop in while loop 3
- c) In while loop 3
- d) Infinite loop

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int i = 0, j = 0;
```

```

    for (i; i < 2; i++){
        for (j = 0; j < 3; j++)
        {
            printf("1\n");
            break;
        }
        printf("2\n");
    }
    printf("after loop\n");
}

```

a)
1
2
after loop

b)
1
after loop

c)
1
2
1
2
after loop

d)
1
1
2
after loop

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int i = 0;
    while (i < 2)
    {
        if (i == 1)
            break;
        i++;
        if (i == 1)
            continue;
        printf("In while loop\n");
    }
    printf("After loop\n");
}

```

a)

- In while loop
- After loop
- b) After loop
- c)
 - In while loop
 - In while loop
 - After loop
- d) In while loop

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int i = 0;
    char c = 'a';
    while (i < 2)
    {
        i++;
        switch (c)
        {
            case 'a':
                printf("%c ", c);
                break;
                break;
        }
    }
    printf("after loop\n");
}
```

- a) a after loop
- b) a a after loop
- c) after loop
- d) error

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("before continue ");
    continue;
    printf("after continue\n");
}
```

- a) Before continue after continue
- b) Before continue
- c) After continue

d) Compile time error

Answer: d

Explanation: None.

Goto & Labels – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("%d ", 1);
    goto l1;
    printf("%d ", 2);
    l1:goto l2;
    printf("%d ", 3);
    l2:printf("%d ", 4);
}
```

- a) 1 4
- b) Compilation error
- c) 1 2 4
- d) 1 3 4

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("%d ", 1);
    l1:l2:
    printf("%d ", 2);
    printf("%d\n", 3);
}
```

- a) Compilation error
- b) 1 2 3
- c) 1 2
- d) 1 3

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
```

```

int main()
{
    printf("%d ", 1);
    goto l1;
    printf("%d ", 2);
}
void foo()
{
    l1 : printf("3 ", 3);
}

```

- a) 1 2 3
- b) 1 3
- c) 1 3 2
- d) Compilation error

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int i = 0, j = 0;
    while (i < 2)
    {
        l1 : i++;
        while (j < 3)
        {
            printf("Loop\n");
            goto l1;
        }
    }
}

```

- a) Loop Loop
- b) Compilation error
- c) Loop Loop Loop Loop
- d) Infinite Loop

Answer: d

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int i = 0, j = 0;
    while (l1: i < 2)
    {
        i++;
        while (j < 3)

```

```

    {
        printf("loop\n");
        goto l1;
    }
}

```

- a) loop loop
- b) Compilation error
- c) loop loop loop loop
- d) Infinite loop

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
    int i = 0, j = 0;
l1: while (i < 2)
{
    i++;
    while (j < 3)
    {
        printf("loop\n");
        goto l1;
    }
}
}

```

- a) loop loop
- b) compilation error
- c) oop loop loop loop
- d) infinite loop

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    int i = 0;
    if (i == 0)
    {
        goto label;
    }
    label: printf("Hello");
}

```

- a) Nothing
- b) Error

- c) Infinite Hello
- d) Hello

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int i = 0, k;
    if (i == 0)
        goto label;
    for (k = 0; k < 3; k++)
    {
        printf("hi\n");
        label: k = printf("%03d", i);
    }
}
```

- a) 0
- b) hi hi hi 0 0 0
- c) 0 hi hi hi 0 0 0
- d) 0 0 0

Answer: a

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int i = 0, k;
    label: printf("%d", i);
    if (i == 0)
        goto label;
}
```

- a) 0
- b) Infinite 0
- c) Nothing
- d) Error

Answer: b

Explanation: None.

Goto & Labels – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int i = 5, k;
    if (i == 0)
        goto label;
    label: printf("%d", i);
    printf("Hey");
}
```

- a) 5
- b) Hey
- c) 5 Hey
- d) Nothing

Answer: c

Explanation: None.

2. goto can be used to jump from main() to within a function.

- a) true
- b) false
- c) depends
- d) varies

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("%d ", 1);
    goto l1;
    printf("%d ", 2);
    l1: goto l2;
    printf("%d ", 3);
    l2: printf("%d ", 4);
}
```

- a) 1 4
- b) Compile time error
- c) 1 2 4
- d) 1 3 4

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("%d ", 1);
    l1:l2:
    printf("%d ", 2);
    printf("%d\n", 3);
}
```

- a) Compile time error
- b) 1 2 3
- c) 1 2
- d) 1 3

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    printf("%d ", 1);
    goto l1;
    printf("%d ", 2);
}
void foo()
{
    l1: printf("3 ", 3);
}
```

- a) 1 2 3
- b) 1 3
- c) 1 3 2
- d) Compile time error

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int i = 0, j = 0;
    while (i < 2)
    {
        l1: i++;
        while (j < 3)
        {
            printf("loop\n");
            goto l1;
        }
    }
}
```

- ```

 }
}
• }

```
- a) loop loop
  - b) Compile time error
  - c) loop loop loop loop
  - d) Infinite loop

**Answer: d**

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int i = 0, j = 0;
 while (l1: i < 2)
 {
 i++;
 while (j < 3)
 {
 printf("loop\n");
 goto l1;
 }
 }
}

```

- a) loop loop
- b) Compile time error
- c) loop loop loop loop
- d) Infinite loop

**Answer: b**

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
int main()
{
 int i = 0, j = 0;
 l1: while (i < 2)
 {
 i++;
 while (j < 3)
 {
 printf("loop\n");
 goto l1;
 }
 }
}

```

- a) loop loop
- b) Compile time error
- c) loop loop loop loop
- d) Infinite loop

**Answer: a**

Explanation: None.

## Basics of Functions – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 void foo();
 printf("1 ");
 foo();
}
void foo()
{
 printf("2 ");
}
```

- a) 1 2
- b) Compile time error
- c) 1 2 1 2
- d) Depends on the compiler

**Answer: a**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 void foo(), f();
 f();
}
void foo()
{
 printf("2 ");
}
void f()
{
 printf("1 ");
 foo();
}
```

- a) Compile time error as foo is local to main
- b) 1 2
- c) 2 1

d) Compile time error due to declaration of functions inside main

**Answer: b**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
 void foo();
 void f()
 {
 foo();
 }
 f();
}
void foo()
{
 printf("2 ");
}
```

- a) 2 2
- b) 2
- c) Compile time error
- d) Depends on the compiler

**Answer: d**

Explanation: Even though the answer is 2, this code will compile fine only with gcc. GNU C supports nesting of functions in C as a language extension whereas standard C compiler doesn't.

4. What will be the output of the following C code?

```
#include <stdio.h>
void foo();
int main()
{
 void foo();
 foo();
 return 0;
}
void foo()
{
 printf("2 ");
}
```

- a) Compile time error
- b) 2
- c) Depends on the compiler

d) Depends on the standard

**Answer: b**

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void foo();
int main()
{
 void foo(int);
 foo(1);
 return 0;
}
void foo(int i)
{
 printf("2 ");
}
```

- a) 2
- b) Compile time error
- c) Depends on the compiler
- d) Depends on the standard

**Answer: a**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void foo();
int main()
{
 void foo(int);
 foo();
 return 0;
}
void foo()
{
 printf("2 ");
}
```

- a) 2
- b) Compile time error
- c) Depends on the compiler
- d) Depends on the standard

**Answer: b**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void m()
{
 printf("hi");
}
void main()
{
 m();
}
```

- a) hi
- b) Run time error
- c) Nothing
- d) Varies

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void m();
void n()
{
 m();
}
void main()
{
 void m()
 {
 printf("hi");
 }
}
```

- a) hi
- b) Compile time error
- c) Nothing
- d) Varies

**Answer: b**

Explanation: None.

## Basics of Functions – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 m();
 void m()
 {
```

```

 printf("hi");
 }
}

```

a) hi  
 b) Compile time error  
 c) Nothing  
 d) Varies

**Answer: b**

Explanation: None.

2. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
 m();
}
void m()
{
 printf("hi");
 m();
}

```

a) Compile time error  
 b) hi  
 c) Infinite hi  
 d) Nothing

**Answer: c**

Explanation: None.

3. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
 static int x = 3;
 x++;
 if (x <= 5)
 {
 printf("hi");
 main();
 }
}

```

a) Run time error  
 b) hi  
 c) Infinite hi  
 d) hi hi

**Answer: d**

Explanation: None.

4. Which of the following is a correct format for declaration of function?

- a) return-type function-name(argument type);
- b) return-type function-name(argument type){}
- c) return-type (argument type)function-name;
- d) all of the mentioned

**Answer: a**

Explanation: None.

5. Which of the following function declaration is illegal?

- a) int 1bhk(int);
- b) int 1bhk(int a);
- c) int 2bhk(int\*, int []);
- d) all of the mentioned

**Answer: d**

Explanation: None.

6. Which function definition will run correctly?

- a)  

```
int sum(int a, int b)
return (a + b);
```
- b)  

```
int sum(int a, int b)
{return (a + b);}
```
- c)  

```
int sum(a, b)
return (a + b);
```
- d) none of the mentioned

**Answer: b**

Explanation: None.

7. Can we use a function as a parameter of another function? [Eg: void wow(int func())].

- a) Yes, and we can use the function value conveniently
- b) Yes, but we call the function again to get the value, not as convenient as in using variable
- c) No, C does not support it
- d) This case is compiler dependent

**Answer: c**

Explanation: None.

8. The value obtained in the function is given back to main by using \_\_\_\_\_ keyword.

- a) return
- b) static
- c) new
- d) volatile



**Answer: a**

Explanation: None

## Functions Returning Non-integers – 1

1. What is the return-type of the function sqrt()?

- a) int
- b) float
- c) double
- d) depends on the data type of the parameter

**Answer: c**

Explanation: None.

2. Which of the following function declaration is illegal?

- a)

```
double func();
int main(){}
double func(){};
```
- b)

```
double func(){};
int main(){};
```
- c)

```
int main()
{
 double func();
}
double func(){}//statements
```
- d) None of the mentioned

**Answer: d**

Explanation: None.

3. What will be the output of the following C code having void return-type function?

```
#include <stdio.h>
void foo()
{
 return 1;
}
void main()
{
 int x = 0;
 x = foo();
 printf("%d", x);
}
```

- a) 1
- b) 0
- c) Runtime error
- d) Compile time error

**Answer: d**

Explanation: None.

4. What will be the data type returned for the following C function?

```
#include <stdio.h>
int func()
{
 return (double)(char)5.0;
}
```

- a) char
- b) int
- c) double
- d) multiple type-casting in return is illegal

**Answer: b**

Explanation: None.

5. What is the problem in the following C declarations?

```
int func(int);
double func(int);
int func(float);
```

- a) A function with same name cannot have different signatures
- b) A function with same name cannot have different return types
- c) A function with same name cannot have different number of parameters
- d) All of the mentioned

**Answer: d**

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 int k = m();
 printf("%d", k);
}
void m()
{
 printf("hello");
}
```

- a) hello 5
- b) Error
- c) Nothing
- d) Junk value

**Answer: a**

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
```

```

int *m()
{
 int *p = 5;
 return p;
}
void main()
{
 int *k = m();
 printf("%d", k);
}

```

- a) 5
- b) Junk value
- c) 0
- d) Error

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
int *m();
void main()
{
 int *k = m();
 printf("hello ");
 printf("%d", k[0]);
}
int *m()
{
 int a[2] = {5, 8};
 return a;
}

```

- a) hello 5 8
- b) hello 5
- c) hello followed by garbage value
- d) Compilation error

**Answer: c**

Explanation: None.

## Functions Returning Non-integers – 2

1. What will be the output of the following C code?

```

#include <stdio.h>
int *m();
void main()
{
 int k = m();
 printf("%d", k);
}

```

```
int *m()
{
 int a[2] = {5, 8};
 return a;
}
```

- a) 5
- b) 8
- c) Nothing
- d) Varies

**Answer: d**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void m(int k)
{
 printf("hi");
}
void m(double k)
{
 printf("hello");
}
void main()
{
 m(3);
}
```

- a) hi
- b) hello
- c) Compile time error
- d) Nothing

**Answer: c**

Explanation: None.

3. What is the default return type if it is not specified in function definition?

- a) void
- b) int
- c) double
- d) short int

**Answer: b**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int foo();
int main()
```

```

{
 int i = foo();
}
foo()
{
 printf("2 ");
 return 2;
}

```

- a) 2
- b) Compile time error
- c) Depends on the compiler
- d) Depends on the standard

**Answer: a**

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
double foo();
int main()
{
 foo();
 return 0;
}
foo()
{
 printf("2 ");
 return 2;
}

```

- a) 2
- b) Compile time error
- c) Depends on the compiler
- d) Depends on the standard

**Answer: b**

Explanation: None.

6. Functions can return structure in C?

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

**Answer: a**

Explanation: None.

7. Functions can return enumeration constants in C?

- a) true
- b) false
- c) depends on the compiler
- d) depends on the standard

**Answer: a**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
enum m{JAN, FEB, MAR};
enum m foo();
int main()
{
 enum m i = foo();
 printf("%d\n", i);
}
int foo()
{
 return JAN;
}
```

- a) Compile time error
- b) 0
- c) Depends on the compiler
- d) Depends on the standard

**Answer: a**

Explanation: None.

## External Variables – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
 m();
 printf("%d", x);
}
int x;
void m()
{
 x = 4;
}
```

- a) 4
- b) Compile time error
- c) 0
- d) Undefined

**Answer: b**

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
int x;
void main()
{
 printf("%d", x);
}
```

- a) Junk value
- b) Run time error
- c) 0
- d) Undefined

**Answer: c**

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int x = 5;
void main()
{
 int x = 3;
 printf("%d", x);
 {
 x = 4;
 }
 printf("%d", x);
}
```

- a) Run time error
- b) 3 3
- c) 3 5
- d) 3 4

**Answer: d**

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int x = 5;
void main()
{
 int x = 3;
 printf("%d", x);
 {
 int x = 4;
 }
 printf("%d", x);
}
```

- a) 3 3
- b) 3 4
- c) 3 5
- d) Run time error

**Answer: a**

Explanation: None.

5. Functions in C are always \_\_\_\_\_

- a) Internal
- b) External
- c) Both Internal and External
- d) External and Internal are not valid terms for functions

**Answer: b**

Explanation: None.

6. Global variables are \_\_\_\_\_

- a) Internal
- b) External
- c) Both Internal and External
- d) None of the mentioned

**Answer: b**

Explanation: None.

7. Which of the following is an external variable in the following C code?

```
#include <stdio.h>
```

```
int func (int a)
```

```
{
```

```
 int b;
```

```
 return b;
```

```
}
```

```
int main()
```

```
{
```

```
 int c;
```

```
 func (c);
```

```
}
```

```
int d;
```

a) a

b) b

c) c

d) d

**Answer: d**

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
 printf("%d", d++);
```

```
}
```



- ```
int d = 10;
```
- a) 9
 - b) 10
 - c) 11
 - d) Compile time error

Answer: d

Explanation: None.

9. What will be the output of the following C code?

```
#include <stdio.h>
double var = 8;
int main()
{
    int var = 5;
    printf("%d", var);
}
```

- a) 5
- b) 8
- c) Compile time error due to wrong format identifier for double
- d) Compile time error due to redeclaration of variable with same name

Answer: a

Explanation: None.

External Variables – 2

1. What will be the output of the following C code?

```
#include <stdio.h>
double i;
int main()
{
    printf("%g\n", i);
    return 0;
}
```

- a) 0
- b) 0.000000
- c) Garbage value
- d) Depends on the compiler

Answer: a

Explanation: None.

2. Which part of the program address space is p stored in the following C code?

```
#include <stdio.h>
int *p = NULL;
int main()
{
    int i = 0;
    p = &i;
```

```

    return 0;
}
a) Code/text segment
b) Data segment
c) Bss segment
d) Stack

```

Answer: b

Explanation: None.

3. Which part of the program address space is p stored in the following C code?

```

#include <stdio.h>
int *p;
int main()
{
    int i = 0;
    p = &i;
    return 0;
}

```

- a) Code/text segment
- b) Data segment
- c) Bss segment
- d) Stack

Answer: c

Explanation: None.

4. Can variable i be accessed by functions in another source file?

```

#include <stdio.h>
int i;
int main()
{
    printf("%d\n", i);
}

```

- a) Yes
- b) No
- c) Only if static keyword is used
- d) Depends on the type of the variable

Answer: a

Explanation: None.

5. Property of the external variable to be accessed by any source file is called by the C90 standard as _____

- a) external linkage
- b) external scope
- c) global scope
- d) global linkage

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
int *i;
int main()
{
    if (i == NULL)
        printf("true\n");
    return 0;
}
```

- a) true
- b) true only if NULL value is 0
- c) Compile time error
- d) Nothing

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
int *i;
int main()
{
    if (i == 0)
        printf("true\n");
    return 0;
}
```

- a) true
- b) true only if NULL value is 0
- c) Compile time error
- d) Nothing

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
static int x = 5;
void main()
{
    x = 9;
    {
        int x = 4;
    }
    printf("%d", x);
}
```

- a) 9
- b) 4
- c) 5
- d) 0

Answer: a

Explanation: None.

Scope of a Variable – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
int i;
int main()
{
    extern int i;
    if (i == 0)
        printf("scope rules\n");
}
```

a) scope rules
b) Compile time error due to multiple declaration
c) Compile time error due to not defining type in statement extern i
d) Nothing will be printed as value of i is not zero because i is an automatic variable

Answer: a

Explanation: None.

2. What will be the output of the following C code (without linking the source file in which ary1 is defined)?

```
#include <stdio.h>
int main()
{
    extern ary1[];
    printf("scope rules\n");
}
```

a) scope rules
b) Linking error due to undefined reference
c) Compile time error because size of array is not provided
d) Compile time error because datatype of array is not provided

Answer: a

Explanation: None.

3. What will be the output of the following C code (after linking to source file having definition of ary1)?

```
#include <stdio.h>
int main()
{
    extern ary1[];
    printf("%d\n", ary1[0]);
}
```

- a) Value of ary1[0];
- b) Compile time error due to multiple definition
- c) Compile time error because size of array is not provided
- d) Compile time error because datatype of array is not provided

Answer: d

Explanation: None.

4. What is the scope of an external variable?

- a) Whole source file in which it is defined
- b) From the point of declaration to the end of the file in which it is defined
- c) Any source file in a program
- d) From the point of declaration to the end of the file being compiled

Answer: d

Explanation: None.

5. What is the scope of a function?

- a) Whole source file in which it is defined
- b) From the point of declaration to the end of the file in which it is defined
- c) Any source file in a program
- d) From the point of declaration to the end of the file being compiled

Answer: d

Explanation: None.

6. Comment on the output of the following C code.

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0; i < 5; i++)
        int a = i;
    printf("%d", a);
}
```

- a) a is out of scope when printf is called
- b) Redclaration of a in same scope throws error
- c) Syntax error in declaration of a
- d) No errors, program will show the output 5

Answer: c

Explanation: None.

7. Which variable has the longest scope in the following C code?

```
#include <stdio.h>
int b;
int main()
{
    int c;
    return 0;
}
int a;
```

- a) a
- b) b
- c) c
- d) Both a and b

Answer: b

Explanation: None.

8. Comment on the following 2 C programs.

```
#include <stdio.h> //Program 1
int main()
{
    int a;
    int b;
    int c;
}
```

```
#include <stdio.h> //Program 2
int main()
{
    int a;
    {
        int b;
    }
    {
        int c;
    }
}
```

- a) Both are same
- b) Scope of c is till the end of the main function in Program 2
- c) In Program 1, variables a, b and c can be used anywhere in the main function whereas in Program 2, variables b and c can be used only inside their respective blocks
- d) None of the mentioned

Answer: c

Explanation: None.

Scope of a Variable – 2

1. What will be the sequence of allocation and deletion of variables in the following C code?

```
#include <stdio.h>
int main()
{
    int a;
    {
        int b;
    }
}
```

- a) a->b, a->b
- b) a->b, b->a

- c) b->a, a->b
- d) b->a, b->a

Answer: b

Explanation: None.

2. Array sizes are optional during array declaration by using _____ keyword.

- a) auto
- b) static
- c) extern
- d) register

Answer: c

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 3;
    {
        x = 4;
        printf("%d", x);
    }
}
```

- a) 4
- b) 3
- c) 0
- d) Undefined

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int x = 5;
void main()
{
    int x = 3;
    m();
    printf("%d", x);
}
void m()
{
    x = 8;
    n();
}
void n()
{
    printf("%d", x);
}
```

- a) 8 3
- b) 3 8
- c) 8 5
- d) 5 3

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int x;
void main()
{
    m();
    printf("%d", x);
}
void m()
{
    x = 4;
}
```

- a) 0
- b) 4
- c) Compile time error
- d) Undefined

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
static int x = 5;
void main()
{
    int x = 9;
    {
        x = 4;
    }
    printf("%d", x);
}
```

- a) 9
- b) 5
- c) 4
- d) 0

Answer: c

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    {
        int x = 8;
    }
    printf("%d", x);
}
```

- a) 8
- b) 0
- c) Undefined
- d) Compile time error

Answer: d

Explanation: None.

Static Variables – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    m();
    m();
}
void m()
{
    static int x = 5;
    x++;
    printf("%d", x);
}
```

- a) 6 7
- b) 6 6
- c) 5 5
- d) 5 6

Answer: a

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    static int x;
    printf("x is %d", x);
}
```

- a) 0
- b) 1
- c) Junk value

d) Run time error

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
static int x;
void main()
{
    int x;
    printf("x is %d", x);
}
```

- a) 0
- b) Junkvalue
- c) Run time error
- d) Nothing

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    static double x;
    int x;
    printf("x is %d", x);
}
```

- a) Nothing
- b) 0
- c) Compile time error
- d) Junkvalue

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    static int x;
    if (x++ < 2)
        main();
}
```

- a) Infinite calls to main
- b) Run time error
- c) Varies
- d) main is called twice

Answer: d

Explanation: None.

6. Which of following is not accepted in C?
- a) static a = 10; //static as
 - b) static int func (int); //parameter as static
 - c) static static int a; //a static variable prefixed with static
 - d) all of the mentioned

Answer: c

Explanation: None.

7. Which of the following cannot be static in C?

- a) Variables
- b) Functions
- c) Structures
- d) None of the mentioned

Answer: d

Explanation: None.

Static Variables – 2

1. What will be the output of the following C code if these two files namely test.c and test1.c are linked and run?

-----file test.c-----

```
#include <stdio.h>
#include ""test.h""
int main()
{
    i = 10;
    printf("""%d """, i);
    foo();
}
```

-----file test1.c-----

```
#include <stdio.h>
#include ""test.h""
int foo()
{
    printf("""%d\n""", i);
}
```

-----file test.h-----

```
#include <stdio.h>
#include <stdlib.h>
static int i;
```

- a) 10 0
- b) 0 0
- c) 10 10
- d) Compilation Error

Answer: a

Explanation: None.

2. Functions have static qualifier for its declaration by default.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

Answer: b

Explanation: None.

3. Is initialisation mandatory for local static variables?

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    foo();
    foo();
}
void foo()
{
    int i = 11;
    printf("%d ", i);
    static int j = 12;
    j = j + 1;
    printf("%d\n", j);
}
```

- a) 11 12 11 12
- b) 11 13 11 14
- c) 11 12 11 13
- d) Compile time error

Answer: b

Explanation: None.

5. Assignment statements assigning value to local static variables are executed only once.

- a) True
- b) False
- c) Depends on the code
- d) None of the mentioned

Answer: b

Explanation: None.

6. What is the format identifier for “static a = 20.5;”?

- a) %s
- b) %d
- c) %f
- d) Illegal declaration due to absence of data type

Answer: b

Explanation: None.

7. Which of the following is true for the static variable?

- a) It can be called from another function
- b) It exists even after the function ends
- c) It can be modified in another function by sending it as a parameter
- d) All of the mentioned

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
void func();
int main()
{
    static int b = 20;
    func();
}
void func()
{
    static int b;
    printf("%d", b);
}
```

- a) Output will be 0
- b) Output will be 20
- c) Output will be a garbage value
- d) Compile time error due to redeclaration of static variable

Answer: a

Explanation: None.

Register Variables – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    register int i = 10;
    int *p = &i;
    *p = 11;
    printf("%d %d\n", i, *p);
}
```

- a) Depends on whether i is actually stored in machine register
- b) 10 10

- c) 11 11
- d) Compile time error

Answer: d

Explanation: None.

2. register keyword mandates compiler to place it in machine register.

- a) True
- b) False
- c) Depends on the standard
- d) None of the mentioned

Answer: b

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    register static int i = 10;
    i = 11;
    printf("%d\n", i);
}
```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    register auto int i = 10;
    i = 11;
    printf("%d\n", i);
}
```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

Answer: b

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
int main()
```

```

{
    register const int i = 10;
    i = 11;
    printf("%d\n", i);
}

```

- a) 10
- b) Compile time error
- c) Undefined behaviour
- d) 11

Answer: b

Explanation: None.

6. Register storage class can be specified to global variables.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

Answer: b

Explanation: None.

7. Which among the following is wrong for "register int a;"?

- a) Compiler generally ignores the request
- b) You cannot take the address of this variable
- c) Access time to a is critical
- d) None of the mentioned

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    register int x = 5;
    m();
    printf("x is %d", x);
}
void m()
{
    x++;
}

```

- a) 6
- b) 5
- c) Junk value
- d) Compile time error

Answer: d

Explanation: None.

Register Variables – 2

1. When compiler accepts the request to use the variable as a register?

- a) It is stored in CPU
- b) It is stored in cache memory
- c) It is stored in main memory
- d) It is stored in secondary memory

Answer: a

Explanation: None.

2. Which data type can be stored in register?

- a) int
- b) long
- c) float
- d) all of the mentioned

Answer: d

Explanation: None.

3. Which of the following operation is not possible in a register variable?

- a) Reading the value into a register variable
- b) Copy the value from a memory variable
- c) Global declaration of register variable
- d) All of the mentioned

Answer: d

Explanation: None.

4. Which among the following is the correct syntax to declare a static variable register?

- a) static register a;
- b) register static a;
- c) Both static register a; and register static a;
- d) We cannot use static and register together

Answer: d

Explanation: None.

5. Register variables reside in _____

- a) stack
- b) registers
- c) heap
- d) main memory

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    register int x = 0;
    if (x < 2)
```



```

{
    x++;
    main();
}

```

- a) Segmentation fault
- b) main is called twice
- c) main is called once
- d) main is called thrice

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
void main()
{
    register int x;
    printf("%d", x);
}

```

- a) 0
- b) Junk value
- c) Compile time error
- d) Nothing

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
register int x;
void main()
{
    printf("%d", x);
}

```

- a) Varies
- b) 0
- c) Junk value
- d) Compile time error

Answer: d

Explanation: None.

Automatic Variables – 1

1. What is the scope of an automatic variable?

- a) Within the block it appears
- b) Within the blocks of the block it appears
- c) Until the end of program
- d) Within the block it appears & Within the blocks of the block it appears

Answer: d

Explanation: None.

2. Automatic variables are allocated space in the form of a _____

- a) stack
- b) queue
- c) priority queue
- d) random

Answer: a

Explanation: None.

3. Which of the following is a storage specifier?

- a) enum
- b) union
- c) auto
- d) volatile

Answer: c

Explanation: None.

4. If storage class is not specified for a local variable, then the default class will be auto.

- a) True
- b) False
- c) Depends on the standard
- d) None of the mentioned

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void foo(auto int i);
int main()
{
    foo(10);
}
void foo(auto int i)
{
    printf("%d\n", i);
}
```

- a) 10
- b) Compile time error
- c) Depends on the standard
- d) None of the mentioned

Answer: b

Explanation: None.

6. Automatic variables are stored in _____

- a) stack
- b) data segment
- c) register

d) heap

Answer: a

Explanation: None.

7. What linkage does automatic variables have?

- a) Internal linkage
- b) External linkage
- c) No linkage
- d) None of the mentioned

Answer: c

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    auto i = 10;
    const auto int *p = &i;
    printf("%d\n", i);
}
```

- a) 10
- b) Compile time error
- c) Depends on the standard
- d) Depends on the compiler

Answer: a

Explanation: None.

Automatic Variables – 2

1. Automatic variables are _____

- a) Declared within the scope of a block, usually a function
- b) Declared outside all functions
- c) Declared with the auto keyword
- d) Declared within the keyword extern

Answer: a

Explanation: None.

2. What is the scope of an automatic variable?

- a) Exist only within that scope in which it is declared
- b) Cease to exist after the block is exited
- c) Exist only within that scope in which it is declared & exist after the block is exited
- d) All of the mentioned

Answer: c

Explanation: None.

3. Automatic variables are allocated memory in _____

- a) heap
- b) Data segment
- c) Code segment

d) stack

Answer: d

Explanation: None.

4. What will be the x in the following C code?

```
#include <stdio.h>
void main()
{
    int x;
}
```

- a) automatic variable
- b) static variable
- c) register variable
- d) global variable

Answer: a

Explanation: None.

5. Automatic variables are initialized to _____

- a) Zero
- b) Junk value
- c) Nothing
- d) Both Zero & Junk value

Answer: b

Explanation: None.

6. Which of the following storage class supports char data type?

- a) register
- b) static
- c) auto
- d) all of the mentioned

Answer: d

Explanation: None.

7. A local variable declaration with no storage class specified is by default _____

- a) auto
- b) extern
- c) static
- d) register

Answer: a

Explanation: None.

C-Preprocessor – 1

1. Property which allows to produce different executable for different platforms in C is called?

- a) File inclusion
- b) Selective inclusion
- c) Conditional compilation

d) Recursive macros

Answer: c

Explanation: Conditional compilation is the preprocessor facility to produce a different executable.

2. What is #include <stdio.h>?

- a) Preprocessor directive
- b) Inclusion directive
- c) File inclusion directive
- d) None of the mentioned

Answer: a

Explanation: None.

3. C preprocessors can have compiler specific features.

- a) True
- b) False
- c) Depends on the standard
- d) Depends on the platform

Answer: a

Explanation: #pragma is compiler specific feature.

4. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(m, n) m * n = 10
int main()
{
    printf("in main\n");
}
```

- a) In main
- b) Compilation error as lvalue is required for the expression m*n=10
- c) Preprocessor error as lvalue is required for the expression m*n=10
- d) None of the mentioned

Answer: a

Explanation: Preprocessor just replaces whatever is given compiler then checks for error at the replaced part of the code. Here it is not replaced anywhere.

Output:

```
$ cc pgm1.c
```

```
$ a.out
```

```
in main
```

5. C preprocessor is conceptually the first step during compilation.

- a) True
- b) False
- c) Depends on the compiler
- d) Depends on the standard

Answer: a

Explanation: None.

6. Preprocessor feature that supply line numbers and filenames to compiler is called?

- a) Selective inclusion
- b) macro substitution
- c) Concatenation
- d) Line control

Answer: d

Explanation: None.

7. #include <somefile.h> are _____ files and #include "somefile.h" _____ files.

- a) Library, Library
- b) Library, user-created header
- c) User-created header, library
- d) They can include all types of file

Answer: d

Explanation: Both of these statement can be used to select any file.

8. What is a preprocessor?

- a) That processes its input data to produce output that is used as input to another program
- b) That is nothing but a loader
- c) That links various source files
- d) All of the mentioned

Answer: a

Explanation: A preprocessor is a program that processes its input data to produce output that is used as input to another program.

C-Preprocessor – 2

1. Which of the following are C preprocessors?

- a) #ifdef
- b) #define
- c) #endif
- d) all of the mentioned

Answer: d

Explanation: None.

2. #include statement must be written _____

- a) Before main()
- b) Before any scanf/printf
- c) After main()
- d) It can be written anywhere

Answer: b

Explanation: Using these directives before main() improves readability.

3. #pragma exit is primarily used for?

- a) Checking memory leaks after exiting the program
- b) Informing Operating System that program has terminated
- c) Running a function at exiting the program
- d) No such preprocessor exist

Answer: c

Explanation: It is primarily used for running a function upon exiting the program.

4. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int one = 1, two = 2;
    #ifdef next
    one = 2;
    two = 1;
    #endif
    printf("%d, %d", one, two);
}
```

- a) 1, 1
- b) 1, 2
- c) 2, 1
- d) 2, 2

Answer: b

Explanation: None.

5. The C-preprocessors are specified with _____ symbol.

- a) #
- b) \$
- c) " "
- d) &

Answer: a

Explanation: The C-preprocessors are specified with # symbol.

6. What is #include directive?

- a) Tells the preprocessor to grab the text of a file and place it directly into the current file
- b) Statements are not typically placed at the top of a program
- c) All of the mentioned
- d) None of the mentioned

Answer: a

Explanation: The #include directive tells the preprocessor to grab the text of a file and place it directly into the current file and are statements are typically placed at the top of a program.

7. The preprocessor provides the ability for _____

- a) The inclusion of header files
- b) The inclusion of macro expansions
- c) Conditional compilation and line control
- d) All of the mentioned

Answer: d

Explanation: The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

8. If #include is used with file name in angular brackets.

- a) The file is searched for in the standard compiler include paths

- b) The search path is expanded to include the current source directory
- c) The search path will expand
- d) None of the mentioned

Answer: a

Explanation: With the #include, if the filename is enclosed within angle brackets, the file is searched for in the standard compiler include paths.

File Inclusion – 1

1. What is the sequence for preprocessor to look for the file within <>?

- a) The predefined location then the current directory
- b) The current directory then the predefined location
- c) The predefined location only
- d) The current directory location

Answer: a

Explanation: <> first searches the predefined location for the specified file and then the current directory.

2. Which directory the compiler first looks for the file when using #include?

- a) Current directory where program is saved
- b) C:COMPILERINCLUDE
- c) S:SOURCEHEADERS
- d) Both C:COMPILERINCLUDE and S:SOURCEHEADERS simultaneously

Answer: b

Explanation: None.

3. What would happen if you create a file stdio.h and use #include "stdio.h"?

- a) The predefined library file will be selected
- b) The user-defined library file will be selected
- c) Both the files will be included
- d) The compiler won't accept the program

Answer: b

Explanation: None.

4. How is search done in #include and #include "somelibrary.h" according to C standard?

- a) When former is used, current directory is searched and when latter is used, standard directory is searched
- b) When former is used, standard directory is searched and when latter is used, current directory is searched
- c) When former is used, search is done in implementation defined manner and when latter is used, current directory is searched
- d) For both, search for 'somelibrary' is done in implementation-defined places

Answer: d

Explanation: None.

5. How is search done in #include and #include "somelibrary.h" normally or conventionally?

- a) When former is used, current directory is searched and when latter is used, standard directory is searched
- b) When former is used, predefined directory is searched and when latter is used, current

directory is searched and then predefined directories are searched

- c) When former is used, search is done in implementation defined manner and latter is used to search current directory
- d) For both, search for some library is done in implementation-defined manner

Answer: b

Explanation: None.

advertisement

6. Can function definition be present in header files?

- a) Yes
- b) No
- c) Depends on the compiler
- d) Depends on the standard

Answer: a

Explanation: None.

7. Comment on the output of the following C code.

```
#include <stdio.h>
#include "test.h"
#include "test.h"
int main()
{
    //some code
}
```

- a) True
- b) Compile time error
- c) False
- d) Depends on the compiler

Answer: b

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(m, n) m ## n
void myfunc();
int main()
{
    myfunc();
}
void myfunc()
{
    printf("%d\n", foo(2, 3));
}
```

- a) 23
- b) 2 3
- c) Compile time error
- d) Undefined behaviour

Answer: a

Explanation: None.

File Inclusion – 2

1. If the file name is enclosed in double quotation marks, then _____

- a) The preprocessor treats it as a user-defined file
- b) The preprocessor treats it as a system-defined file
- c) The preprocessor treats it as a user-defined file & system-defined file
- d) None of the mentioned

Answer: a

Explanation: None.

2. If the file name is enclosed in angle brackets, then _____

- a) The preprocessor treats it as a user-defined file
- b) The preprocessor treats it as a system-defined file
- c) The preprocessor treats it as a user-defined file & system-defined file
- d) None of the mentioned

Answer: b

Explanation: None.

3. What will be the output of the following C code snippet?

```
#include (stdio.h)
void main()
{
    printf("hello");
}
```

- a) hello
- b) Nothing
- c) Compile time error
- d) Depends on compiler

Answer: c

Explanation: File to be included must be specified either in "" or <>.

Output:

```
$ cc pgm1.c
```

```
pgm1.c:1: error: #include expects "FILENAME" or
```

```
pgm1.c: In function 'main':
```

```
pgm1.c:4: warning: incompatible implicit declaration of built-in function 'printf'
```

4. The below two lines are equivalent to _____

```
#define C_IO_HEADER <stdio.h>
#include C_IO_HEADER
```

- a) #include<stdlib.h>
- b) #include"printf"
- c) #include"C_IO_HEADER"

d) #include<stdio.h>

Answer: d

Explanation: Since C_IO_HEADER is defined to be <stdio.h>, the second line becomes #include<stdio.h>, since C_IO_HEADER is replaced with <stdio.h>

5. What will be the output of the following C code?

```
#include <stdio.h>
#include "printf"
void main()
{
    printf("hello");
}
```

- a) hello
- b) Error
- c) Depends on compiler
- d) Varies

Answer: a

Explanation: None.

6. Which of the following file extensions are accepted with #include?

- a) .h
- b) .in
- c) .com
- d) All of the mentioned

Answer: d

Explanation: The preprocessor will include whatever file extension you specify in your #include statement. However, it is not a good practice as another person debugging it will find it difficult in finding files you have included.

7. Which of the following names for files not accepted?

- a) header.h.h
- b) 123header.h
- c) _head_er.h
- d) None of the mentioned

Answer: d

Explanation: All file names are accepted as for the execution to occur. There are no constraints on giving file names for inclusion.

Macro Substitution – 1

1. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(m, n) m ## n
int main()
{
    printf("%s\n", foo(k, l));
}
```

- a) k l
- b) kl

- c) Compile time error
- d) Undefined behaviour

Answer: c

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(m, n) " m ## n "
int main()
{
    printf("%s\n", foo(k, l));
}
```

- a) k l
- b) kl
- c) Compile time error
- d) m ## n

Answer: d

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(x, y) #x #y
int main()
{
    printf("%s\n", foo(k, l));
    return 0;
}
```

- a) kl
- b) k l
- c) xy
- d) Compile time error

Answer: a

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d\n", foo(i + j, 3));
    return 0;
}
```

- a) Divided by zero exception
- b) Compile time error
- c) -8

d) -4

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
void f();
int main()
{
    #define foo(x, y) x / y + x
    f();
}
void f()
{
    printf("%d\n", foo(-3, 3));
}
```

a) -8

b) -4

c) Compile time error

d) Undefined behaviour

Answer: b

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
void f();
int main()
{
    #define max 10
    f();
    return 0;
}
void f()
{
    printf("%d\n", max * 10);
}
```

a) 100

b) Compile time error since #define cannot be inside functions

c) Compile time error since max is not visible in f()

d) Undefined behaviour

Answer: a

Explanation: None.

7. What will be the output of the following C code?

```
#include <stdio.h>
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
```

```

printf("%d ", foo(i + j, 3));
printf("%d\n", foo(-3, 3));
return 0;
}

```

- a) -8 -4
- b) -4 divided by zero exception
- c) -4 -4
- d) Divided by zero exception

Answer: a

Explanation: None.

8. What will be the output of the following C code?

```

#include <stdio.h>
int foo(int, int);
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d ",foo(i + j, 3));
    #undef foo
    printf("%d\n",foo(i + j, 3));
}
int foo(int x, int y)
{
    return x / y + x;
}

```

- a) -8 -4
- b) Compile time error
- c) -8 -8
- d) Undefined behaviour

Answer: a

Explanation: None.

9. What is the advantage of #define over const?

- a) Data type is flexible
- b) Can have a pointer
- c) Reduction in the size of the program
- d) None of the mentioned

Answer: a

Explanation: None.

Macro Substitution – 2

1. What will be the output of the following C code?

```

1.      #include <stdio.h>
2.      void main()
3.      {
4.          #define max 37;
5.          printf("%d", max);

```

6. }
- a) 37
 - b) Compile time error
 - c) Varies
 - d) Depends on compiler

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    #define max 37
    printf("%d", max);
}
```

- a) 37
- b) Run time error
- c) Varies
- d) Depends on compiler

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    #define const int
    const max = 32;
    printf("%d", max);
}
```

- a) Run time error
- b) 32
- c) int
- d) const

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    #define max 45
    max = 32;
    printf("%d", max);
}
```

- a) 32
- b) 45
- c) Compile time error
- d) Varies

Answer: c

Explanation: None.

5. What will be the output of the following C code?

```
#include <stdio.h>
# define max
void m()
{
    printf("hi");
}
void main()
{
    max;
    m();
}
```

- a) Run time error
- b) hi hi
- c) Nothing
- d) hi

Answer: d

Explanation: None.

6. What will be the output of the following C code?

```
#include <stdio.h>
#define A 1 + 2
#define B 3 + 4
int main()
{
    int var = A * B;
    printf("%d\n", var);
}
```

- a) 9
- b) 11
- c) 12
- d) 21

Answer: b

Explanation: None.

7. Which of the following Macro substitution are accepted in C?

- a)


```
#define A #define
A VAR 20
```
- b)


```
#define A define
#A VAR 20
```
- c)


```
#define #A #define  
#A VAR 20
```

d) None of the mentioned

Answer: d

Explanation: None.

8. Comment on the output of the following C code.

```
#include <stdio.h>  
#define var 20);  
int main()  
{  
    printf("%d\n", var  
}
```

- a) No errors, it will show the output 20
- b) Compile time error, the printf braces aren't closed
- c) Compile time error, there are no open braces in #define
- d) None of the mentioned

Answer: a

Explanation: None.

9. Which of the following properties of #define is not true?

- a) You can use a pointer to #define
- b) #define can be made externally available
- c) They obey scope rules
- d) All of the mentioned

Answer: d

Explanation: None.

Conditional Inclusion – 1

1. What will be the output of the following C code?

```
#include <stdio.h>  
#define SYSTEM 20  
int main()  
{  
    int a = 20;  
    #if SYSTEM == a  
    printf("HELLO ");  
    #endif  
    #if SYSTEM == 20  
    printf("WORLD\n");  
    #endif  
}
```

- a) HELLO
- b) WORLD
- c) HELLO WORLD
- d) No Output

Answer: b

Explanation: None.

2. What will be the output of the following C code?

```
1.  #include <stdio.h>
2.  #define Cprog
3.  int main()
4.  {
5.      int a = 2;
6.      #ifdef Cprog
7.          a = 1;
8.          printf("%d", Cprog);
9.      }
```

- a) No output on execution
- b) Output as 1
- c) Output as 2
- d) Compile time error

Answer: d

Explanation: None.

3. The “else if” in conditional inclusion is written by?

- a) #else if
- b) #elseif
- c) #elsif
- d) #elif

Answer: d

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
#define COLD
int main()
{
    #ifdef COLD
    printf("COLD\t");
    #undef COLD
    #endif
    #ifdef COLD
    printf("HOT\t");
    #endif
}
```

- a) HOT
- b) COLD
- c) COLD HOT
- d) No Output

Answer: b

Explanation: None.

5. Which of the following sequences are unaccepted in C language?

- a)
#if
#else
#endif
- b)
#if
#elif
#endif
- c)
#if
#if
#endif
- d)
#if
#undef
#endif

Answer: c

Explanation: None.

6. In a conditional inclusion, if the condition that comes after the if is true, then what will happen during compilation?

- a) Then the code up to the following #else or #elif or #endif is compiled
- b) Then the code up to the following #endif is compiled even if #else or #elif is present
- c) Then the code up to the following #elif is compiled
- d) None of the mentioned

Answer: a

Explanation: None.

7. Conditional inclusion can be used for _____

- a) Preventing multiple declarations of a variable
- b) Check for existence of a variable and doing something if it exists
- c) Preventing multiple declarations of same function
- d) All of the mentioned

Answer: d

Explanation: None.

8. The #elif directive cannot appear after the preprocessor #else directive.

- a) True
- b) False

Answer: a

Explanation: None.

Conditional Inclusion – 2

1. For each #if, #ifdef, and #ifndef directive.

- a) There are zero or more #elif directives
- b) Zero or one #else directive
- c) One matching #endif directive
- d) All of the mentioned

Answer: d

Explanation: None.

2. The #else directive is used for _____

- a) Conditionally include source text if the previous #if, #ifdef, #ifndef, or #elif test fails
- b) Conditionally include source text if a macro name is not defined
- c) Conditionally include source text if a macro name is defined
- d) Ending conditional text

Answer: a

Explanation: None.

3. What will be the output of the following C code?

```
#include <stdio.h>
#define MIN 0
#if MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

- a) 10 0
- b) Compile time error
- c) Undefined behaviour
- d) None of the mentioned

Answer: b

Explanation: None.

4. What will be the output of the following C code?

```
#include <stdio.h>
#define MIN 0
#ifdef MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
}
```

```

    return 0;
}
a) 10 0
b) Compile time error
c) Undefined behaviour
d) None of the mentioned

```

Answer: a

Explanation: None.

5. What will be the output of the following C code?

```

#include <stdio.h>
#define MIN 0
#if defined(MIN) + defined(MAX)
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
a) 10 0
b) Compile time error
c) Undefined behaviour
d) Somegarbagevalue 0

```

Answer: a

Explanation: None.

6. What will be the output of the following C code?

```

#include <stdio.h>
#define MIN 0
#if defined(MIN) - (!defined(MAX))
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
a) 10 0
b) Compile time error
c) Undefined behaviour
d) Somegarbagevalue 0

```

Answer: b

Explanation: None.

7. What will be the output of the following C code?

```

#include <stdio.h>
#define MIN 0
#ifdef(MIN)

```

```
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

- a) 10 0
- b) Compile time error
- c) Run time error
- d) Preprocessor error

Answer: d

Explanation: None.

8. What will be the output of the following C code?

```
#include <stdio.h>
#define MIN 0;
#ifdef MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN
    return 0;
}
```

- a) 10 0
- b) Compile time error due to illegal syntax for printf
- c) Undefined behaviour
- d) Compile time error due to illegal MIN value

Answer: a

Explanation: None