1. The precedence of arithmetic operators is (from highest to lowest)

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

    Solution: (a) The precedence order follows the first option (a)

2. Which of the following operator takes only integer operands?
    a) +
    b) /
    c) *
    d) %

Solution: (d) modulo (%) operation only takes integer

3. Which of the following method are accepted for assignment?
    a) 8=x=y=z
    b) x=8=y=z
    c) x=y=z=8
    d) None

Solution: (c)

4. Which of the following statement is correct?
    a) Operator precedence determines which operator is performed first in an expression with more than one operator with different precedence. Associativity is used when two operators of same precedence appear in an expression
    b) Operator associativity determines which operator is performed first in an expression with more than one operator with different associativity. Precedence is used when two operators of same precedence appear in an expression
    c) Operator precedence and associativity are same.
    d) None of the above

Solution: (a) Operator precedence determines which operator is performed first in an expression with more than one operator with different precedence, whereas associativity is used when two operators of same precedence appear in an expression

5. What is the output of the following program?

```
#include <stdio.h>
main()
 {
        float i = -3.0;
        int k = i % 2;
        printf("%d", k);
}
```

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

Solution: (d) Modulus operator, '%' cannot be operated on floating variable. Thus i%2 is not a valid operation in C. The compiler will show error at this step.

6. Compute the value of x of the C program given below

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

a) 17.33
b) 14
c) 17
d) Depends on compiler

Solution: (c) The operator * and / have same precedence and their associativity is from left to right so first * is evaluated before / and + operator have lower precedence than these two so it is evaluated at last. As x and all the numbers are integer so only integer part will be considered. Thus the correct output is 17

7. What is the output of the following program?

```
#include<stdio.h>
main()
{
  int a=10, b=3, c;
  float d;
  c=a%b;
  d=a/b;
  printf("Value of c and d are %d and %f respectively",c, d);
}
```

a) Value of c and d are 1 and 3 respectively
b) Value of c and d are 1 and 3.333333 respectively
c) Value of c and d are 1.000000 and 3.000000 respectively
d) Value of c and d are 1 and 3.000000 respectively

Solution: (d) Modulo (%) gives the reminder and division (/) gives the quotient of a division operation. Here, d is a floating variable and a and b are both integers. Thus c and d would be 1 and 3.000000 respectively.

8. Find the output of the following C code

```
#include<stdio.h>
main()
{
   int a=10, b=3, c=2, d=4, result;
   result=a+a*-b/c%d+c*d;
   printf("%d", result);
}
```

a) -42
b) 24
c) 15
d) -34

Solution: (c) Following the precedence rule, we can conclude that the operation steps are

- ➔ Result=10+10*- 3/2%4+2*4
- ➔ Result=10-30/2%4+2*4
- ➔ Result=10-15%4+2*4
- ➔ Result=10-3+2*4
- ➔ Result=10-3+8
- ➔ Result=7+8
- ➔ Result=15

9. What is the output of the following C code?

```
#include <stdio.h>
main()
{

   int y = 2;
   int z = y +(y == 10);
   printf("%d\n", z);
}
```

a) 12
b) 2
c) 20
d) Compiler error

Solution: (b) '==' is a relational operator. It returns 1 if the condition is true or 0 if the condition is false. Thus y==10 will return false (which is 0) and 2+0 will be 2. Hence, the value of z will be 2.

10. The output of the following program will be

```
#include<stdio.h>
main()
{
        int a=0,b=1,c=-1;
        if(a)
        printf("IITKGP \n");
        if(b)
        printf("IITM \n");
        if(c)
        printf("IITR \n");
}
```

a) IITKGP
b) IITM IITR
c) IITM
   IITR
d) IITKGP
   IITR

Solution: (c) +1 and -1 is taken as true inside the if statement. Whereas 0 will be considered as false. Thus the printf corresponding to b and c will be printed.

11. What is the output of the following program?

```
#include<stdio.h>
int main()
{
   int i;
   if(i=0,2,3)
      printf("NPTEL ");
   else
      printf("Programming on C ");
   printf("%d\n",i);
}
```

a) Programming on C 0
b) NPTEL 0
c) NPTEL 3
d) Compilation error

Solution: (b) At first zero will assign in ‘i’ then comma operator returns the last value which is 3 and condition becomes true.

12. Find the output of the following program.

```
#include <stdio.h>
main()
{
   int x = 5;
   if (x>=6);
   printf("hello");
}
```

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

Solution: (a) The condition inside the if statement is false. The line is ended after if statement. Thus the compilation comes to the next line and hello will be printed. Thus the program prints “hello”.

13. What will be the output?

```
#include<stdio.h>
int main()
{
   int a;
   a = 10, 20, 30;
   printf("%d", a);
   return 0;
}
```

a) 10
b) 20
c) 30
d) Compilation error

Solution: (a) a = 10, 20, 30; will be evaluated as (a = 10), 20, 30 since comma has the least precedence among all operators.

14. What will be the output?

```
#include <stdio.h>
int main()
{
   int a = 100, b = 200, c = 300;
   if (c > b > a)
         printf("TRUE");
   else
         printf("FALSE");
   return 0;
```

```
}
```

a) TRUE
b) FALSE
c) Syntax Error
d) Compilation Error

Solution: (b) FALSE :: (c > b > a) is treated as ((c > b) > a), associativity of '>' is left to right. Therefore the value becomes ((300 > 200) > 100) which becomes (1 > 100) thus FALSE

15. What will be the output?

```
#include<stdio.h>
int main()
{
  int x;
  x= 5>7+3 && 8;
  printf("%d", x);
  return 0;
}
```

a) 0
b) 1
c) 9
d) Compilation error

Solution: (a) 0 :: This expression is equivalent to:

((5> (7 + 3)) && 8)

i.e, (7 + 3) executes first resulting into 10

then, first part of the expression (5 > 10) executes resulting into 0 (false)

Then, (0 && 8) executes resulting into 0 (false)

16. What will be the output?

```
#include<stdio.h>
int main()
{
  int x;
  x= 10==20!=30;
  printf("%d", x);
  return 0;
}
```

a) 0
b) 1
c) 10
d) 30

Solution: (b) 1

Here, operators == and != have same precedence. The associativity of both == and != is left to right, i.e, the expression on the left is executed first and moves towards the right.

Thus, the expression above is equivalent to :

((10 == 20) != 30)

i.e, (10 == 20) executes first resulting into 0 (false)

then, (0 != 30) executes resulting into 1 (true)

17. What will be the output?

```
#include <stdio.h>
int main()
{
   int x=2;
   if(x=1)
      printf("TRUE");
   else
      printf("FALSE");
   return 0;
}
```

   a) TRUE
   b) FALSE
   c) Compilation Error
   d) Compiler Dependent

Solution: (a) TRUE
TRUE
if(x=1)... "=" is an assignment operator, so 1 will be assigned to x and condition will be true due to if(1).

18. What will be the output?

```
#include <stdio.h>
int main()
{
   if( (-10 && 10)||(20 && -20) )
      printf("Condition is true.");
   else
      printf("Condition is false.");
   return 0;
}
```

   a) Condition is true
   b) Condition is false
   c) Error
   d) No output possible

Solution: (a) Condition is true

Any non-zero value is treated as true for condition. Consider the expressions: if((-10 && 10)||(20 && -20))
=if((1) || (1))
=if(1)

19. What will be the output?

```
#include <stdio.h>
#define TRUE 1
int main()
{
   if(TRUE)
      printf("1");
      printf("2");
   else
      printf("3");
      printf("4");
   return 0;
}
```

a) 1 2 3 4
b) 1 2
c) Compilation error
d) 3 4

Solution: (c) Compilation error

Illegal else without matching if. You can use only one statement within the if( )without parenthesis {...} .

20. Which of the following are incorrect statements?
If int a=7.

1) if(a==7) printf("IncludeHelp");
2) if(7==a) printf("IncludeHelp");
3) if(a=7) printf("IncludeHelp");
4) if(7=a) printf("IncludeHelp");

a) 1 and 2.
b) 3 only.
c) 4 only.
d) 2,3 and 4.

Solution: (c) 4 only.
Consider the following expressions:
if(a==7) => if(1) => correct.
if(7==a) => if(1) => correct.
if(a=7) => if(7) => correct (7 is a non-zero value).
if(7=a) => incorrect, because value of ‘a’ cannot assign in 7, L value required error will occur.