Week-3 Assignment Solution

1. The precedence (from highest to lowest) of arithmetic operators is
   a) %, *, /, +, –
   b) %, +, /, *, –
   c) +, -, %, *, /
   d) %, +, -, *, /

Solution: (a) The precedence order follows the first option (a). %, *, /, have the same precedence and is associated from left to right. Then +, –.

2. Which of the following are incorrect statements?
   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==10) => if(1) => correct.
if(10==a) => if(1) => correct.
if(a=10) => if(10) => correct (10 is a non-zero value).
if(10=a) => incorrect, because value of a cannot assign in 10, Lvalue required error will occur.

3. 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.

4. Find the output of the following C code

```
#include<stdio.h>
int main()
{
  int a=12, b=18, c=7, d=4, result;
  result=a+a*-b/c%d+c*d;
  printf("%d", result);
  return 0;
}
```

a) -18
b) 38
c) 24
d) -32

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

➔ Result=12+12*-18/7%4+7*4
➔ Result=12-216/7%4+7*4
➔ Result=12-30%4+7*4
➔ Result=12-2+7*4
➔ Result=12-2+28
➔ Result=10+28
➔ Result=38

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

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

   int y = 10;
   int z = y +(y == 10);
   printf("%d\n", z);
   return 0;
}
```

a) 10

b) 11
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 true (which is 1) and 10+1 will be 11. Hence, the value of z will be 11.

6. What will be the output?

```
#include <stdio.h>
int main()
{
   int h = 9;
   int b = 5 * 2 + 2 * 3 < h*2 ?  5 : 3;
   printf("%d", b);
   return 0;
}
```

Solution: 5 (short answer type)

7. What is the output of the C program given below?

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

a) true
b) false
c) Compiler dependent
d) Compiler error

Solution: (b) ++ is a post increment operator. In x++, first 0 will be assigned and then x will be incremented by 1. Thus the next else if condition will be evaluated and false will be printed.

8. What will be the output?

```
#include <stdio.h>
int main()
{
   int a = 1, b = 2, c = 3;
   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 ((3 > 2) > 1) which becomes (1 > 1) thus false

9. 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

10. What if the output of the program below?

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

```
int main()
{
   int a = 1, b = -1, c = 0, d;
   d = ++a && ++b || c--;
   if (d)
        printf("Kolkata \n");
   else if(c)
        printf("Delhi \n");
   else
        printf("Bangalore \n");
   return 0;
}
```

a) Kolkata
b) Delhi
c) Bangalore
d) Compiler error

Solution: (b) While computing the value of d we get ++a=2, ++b=0 and c--=0. Thus the value of d is 0 according to the logical operation. Later the value of c is decremented by 1 as -- is a post-increment operator. Thus, else if condition will be evaluated and Delhi will be printed.