Problem Solving through Programming in C
Week 03 Assignment Solution

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 is a logical operator in C?
   a) &&
   b) ==
   c) =
   d) +=

Solution: (a) The && operator is a logical AND operator in C.

3. What is the output of the following program?

```
#include <stdio.h>
int main()
 {
    float i = -3.0;
    int k = i % 2;
    printf("%d", k);
        return 0;
}
```

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

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

4. Find the output of the following C code. (% indicates modulo operation, which results in the remainder of a division operation)

```
#include<stdio.h>
int main()
{
  int a=10, b=3, c=2, d=4, result;
  result=a+a*-b/c%d+c*d;
  printf("%d", result);
  return 0;
}
```

   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

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

```
#include <stdio.h>
int main()
{
int h = 8;
int b = 4 * 6 + 3 * 4 < h*5 ?4 : 3;
printf("%d\n", b);
return 0;
}
```

a) 0
b) 3
c) 4
d) Compilation error

Solution: (c) '?:' is Conditional Expression. If Condition is true ? then value X : otherwise value Y. After simplifying the expression, we get 36<40?4:3. The condition in LHS of ? is true. Thus 4 will be stored in b.

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

```
#include <stdio.h>
int main() {
   int x = 1, y = 0;
   if (x && y) {
      printf("Both are true\n");
   } else {
      printf("At least one is false\n");
   }
   return 0;
}
```

a) Both are true
b) At least one is false
c) Compilation error
d) None of the above

Solution: (b) Since y is 0, the condition x && y evaluates to false, so "At least one is false" will be printed.

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

```
#include <stdio.h>
int main() {
   int x = 10;
   if (x = 5) {
      printf("x is 5\n");
   } else {
      printf("x is not 5\n");
   }
   return 0;
}
```

a) x is 5
b) x is not 5
c) Compilation error
d) None of the above

Solution: (a) The condition if (x = 5) is an assignment, not a comparison. It assigns 5 to x, which evaluates to true, so "x is 5" will be printed.

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

```
#include <stdio.h>
int main() {
   int a = 10, b = 5;
   if (a > b)
      if (b > 0)
         printf("b is positive\n");
      else
         printf("b is non-positive\n");
   printf("a is greater than b\n");
   return 0;
}
```

a) b is positive
b) b is non-positive
c) a is greater than b
d) b is positive a is greater than b

Solution: (d) Both "b is positive" and "a is greater than b" will be printed because the second printf is not part of the inner if-else block.

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

Solution: (c)

10. What will be the output?

```
#include<stdio.h>
int main()
{
int x;
  x= 9<5+3 && 7;
printf("%d", x);
return 0;
}
```

a) 0
b) 1
c) 7
d) Compilation error

Solution: (a) 0

This expression is equivalent to:
((9< (5 + 3)) && 7)
i.e., (5 + 3) executes first resulting into 8
then, the first part of the expression (9< 8) executes, resulting in 0 (FALSE)
Then, (0 && 7) execute resulting into 0 (FALSE)