Exercise 6: Functions

1. What are the advantages of writing modular programs?

2. Fill in the blanks.

(a)

```
_____________________;      //-- function prototype
int main()
{
   cout << "Laugh out loud 5 times.\n";
   laugh();
   return 0;
}

void laugh(void)
{
   for (int i=0;i<5;i++)
     cout << "LOL\n";
}
```

Sample console output:

```
Laugh out loud 5 times.
LOL
LOL
LOL
LOL
LOL
```

(b)

```
_____________________;      //-- function prototype
int main()
{
   int x;
   cout << "Enter a number : ";
   cin >> x;
   cout << "Laugh out loud " << x << " times.\n";

   ___________________;    //-- call the function
   return 0;
}


void laugh(int _______)
{
   for (int i=0;i<num;i++)
     cout << "LOL\n";
}
```

Sample console output#1:

```
Enter a number: 3
Laugh out loud 3 times.
LOL
LOL
LOL
```

Sample console output#2:

```
Enter a number: 6
Laugh out loud 6 times.
LOL
LOL
LOL
LOL
LOL
LOL
```

(c) The function **validateMark** checks if the number entered by the user is within 0 – 100.

```
________________________;     //-- function prototype
int main()
{
   int x;
   cout << "Enter a number : ";
   cin >> x;
   if (validateMark(____)== _____ )
      cout << "Invalid marks.\n";
   else
      cout << "Thank U.\n";
   return 0;
}


_____ validateMark (___________)
{
   if (num>=0 && num<=100)
      return 1;
   else return 0;
}
```

Sample console output#1:

```
Enter a number: 3
Thank U.
```

Sample console output#2:

```
Enter a number: -5
Invalid marks.
```

Sample console output#3:

```
Enter a number: 88
Thank U.
```

Sample console output#4:

```
Enter a number: 101
Invalid marks.
```

3. The following program has some syntax and a logical error, because of this it gives an incorrect output. Spot and correct the errors.

```
int displaySum(void);     //-- function prototype
int sum;                  //-- global variable
int main()
{
   int num1, num2;
   cout << "Enter a number : ";
   cin >> num1;
   cout >> "Enter another number : ";
   cin >> num2;
   sum = num1 + num1;
   displaySum()
   return 0;
}

void displaySum(void);
{
   int sum;
   cout << "The sum is : " << sum <"\n";
   return;
}
```

4. Write a program that will analyze, for your class, the grades obtained by all the students for Structured Programming. The program will prompt the user to enter the grade for each student. Valid grades are A, B, and C. The program calculates and displays the total number of As, Bs and Cs. The user should be able to enter the grades in uppercase or lowercase. You may assume there are only 10 students in your class.

   Your program must be modular. Write a function to read and total the grades and another to print the results. A skeleton of the program is given below:

```
char grade; //-- global variables
int totalA, totalB, totalC;

int main()
{
    readandTotalGrades();
    displayTotals();
    return 0;
}
```

Sample console output:

```
Please enter grade for student 1:A
Please enter grade for student 2:B
Please enter grade for student 3:C
Please enter grade for student 4:A
Please enter grade for student 5:B
Please enter grade for student 6:C
Please enter grade for student 7:C
Please enter grade for student 8:C
Please enter grade for student 9:C
Please enter grade for student 10:C
Total no. of grade A students : 2
Total no. of grade B students : 2
Total no. of grade C students : 6
```

5a. Write a program, which prompts the user to enter three integer numbers. It then finds and displays the smallest of the three numbers. The program outline is:

```
int main()
{
  int num1, num2, num3, smallest;

  cout << "Enter the first number : ";
  cin >> num1;
  cout << "Enter the second number : ";
  cin >> num2;
  cout << "Enter the third number : ";
  cin >> num3;
  smallest = findSmallest(num1, num2, num3);
  cout << "The smallest number is : " << smallest;
  return 0;
}
```

Sample console output:

```
Enter the first number : 200
Enter the second number : -2
Enter the third number : 8
The smallest number is : -2
```

5b. Create two more functions, similar to the function *findSmallest()*, one to find the largest and another to find the average of the three numbers. The function prototypes are as follows:

```
int findLargest(int, int, int);
double findAverage(int, int, int);
```

Sample console output:

```
Enter the first number : 1
Enter the second number : -20
Enter the third number : 100
The smallest number is : -20
The largest number is : 100
The average is : 27
```