Sukkur IBA University
Department of Computer Science

FUNDAMENTALS OF PROGRAMMING
LAB 08 - FUNCTIONS/BEHAVIORS

Saif Hassan

Date: 21st October, 2019

READ IT FIRST

Prior to start solving the problems in this assignments, please give full concentration on following points.

1. WORKING - This is individual lab. If you are stuck in a problem contact your teacher, but, in mean time start doing next question (don't waste time).

2. DEADLINE - Before Next Lab.

3. SUBMISSION - This assignment needs to be submitted in a soft copy.

4. WHERE TO SUBMIT - Please visit your Google Classroom.

5. HOW TO SUBMIT - rename all .cpp file as 000-00-0000_lab08.cpp, and submit .cpp file ONLY.

KEEP IT WITH YOU!

1. Indent your code inside the main () function. It's a good practice!

2. It is not bad if you keep your code indented inside the loops, if and else blocks as well.

3. Comment your code, where it is necessary.

4. Read the entire question. Don't jump to the formula directly.

Exercises

You can answer all **functions** asked in this section in a single **.cpp** file and test them in **main ( )** function.

1. Write a **function** that will take **3 arguments** as an input i.e. **acceleration**, **time** and **initial velocity**, and return the **distance** covered. The formula of distance is $S = V_i + 0.5 * a * t^2$. The prototype of the function is given as: **float distance (float acc, float time, float initial)**. You can test in main function as:

```
1. float s = distance(60, 1.5, 10);
2. cout<<s;
```

2. Write a **function** with the name **printFancy** which takes **two** input **params** i.e. a **character x** and an **integer n**. You have to print **x** up to **n times**. For example, if I use: **printFancy ('*', 10);** it will print on screen * **10 times** as ************.** You can test in main function as: **printFancy ('*', 10);**

3. What should be done with the **function** above to give space after each character? For example, if I use: **printFancy2 ('-', 10);** it will print on screen - **10 times** as **- - - - - - - - - -.**

4. Write a **function** that will take **two params** as an integer input **i.e start** and **end** and **return** the **sum** of all numbers existing in between.

5. Write a **function** that will take **three params** as an **integer** input **i.e start**, **end** and **n**. And then print the numbers that are **divisible** by **n** in the range defined by **start** and **end**.

6. Write a **function** named **swap(x, y)** where **x** and **y** are **integers**.

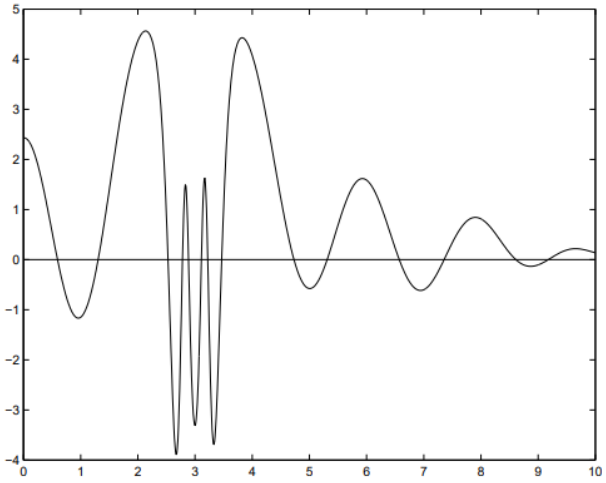
7. Write a **function** that take **three inputs**: **x, y** and **choice** and **return** value based on user's **choice**. Choice may be: (S → SQUARE, C → CUBIC, ^ → POWER, M → MAX, N → MIN, + → ADDITION, * → MULTIPLICATION).

8. Write following math functions, calculate formula and return value:

$$f(x) = 0.1 + \cos(3x) + 3\sin\left(\frac{1}{0.1 + (x-3)^2}\right) + \cos(\sqrt{11}\,x)$$

You may use **cmath** library for sin/cos functions.

Note: You can test f(x) *on x-axis in given graph.*



9. Write a **function** that takes **one param** as **integer** input **taskNo** and calls function accordingly. Let's say user's input is **6**, then ask user input **x, y** then **call swap(x, y)** function. The prototype of the function is given as: **void execTask (int taskNo)**

10. Write a **void** function named **showMenu()** in which you have to show user, all the tasks number and name in proper manner. Output may be:

```
1- Calculate Distance
2- Fancy
3- Improved Fancy
4- Sum of all numbers in given range
and so on.
```

Now ask user any **input** to call **execute** that task.

11. Code for **main ()** function is as follows:

```
1. // Do Not Change main function
2. int main ()
3. {
4.     int choice;
5.
6.     showMenu()
7.     cin>>choice;
8.     execTask(int taskNo);
9.
10.    system ("PAUSE") ;
11.    return 0 ;
12.}
```