

# Lab07

## 1. Lab Topics

This lab covers different topics, but more emphasis is placed on functions and modularization.

## 2. Teacher/Student Portal

**REMEMBER LAB06.** This lab builds on that in which some parts of the code remain the same but now you are adding more functions to the program.

You are tasked with creating a teacher/student portal with C++. This portal helps a teacher calculate and display statistics for a class of students. The program will allow the teacher to enter student data and then provide statistics like the average score, highest score, lowest score, and the number of students who passed or failed. The other part of the portal helps students to perform basic mathematical operations. Your task is to create this program using functions for better organization and readability. Follow the steps below:

### Step 1:

Implement the following “new” functions:

1. Implement a function called **getUserName()** that returns the name of the system user.
2. Implement a function called **getUserID()** that returns the ID of the system user. Ensure that this ID is exactly 5 characters in length. You can reuse your code from **lab05** that required a 5-character sequence input.
3. Implement a function called **calculateASCIISum(getUserID())** that calculates the ASCII sum of the first 3 characters of the 5-character sequence entered in part 2. Return the sum. This function should take the **getUserID()** function as an argument.
4. Define a greeting function called **greetUser()** that takes two arguments: the result of the function defined in part 3 and the result of the function defined in part 1. In this function:
  - Check if the return value of the function from part 3 is strictly less than 150.
  - If yes, output "Welcome student [name]".
  - If no, output "Welcome teacher [name]".
  - Also, output the current timestamp in a readable format, such as "You logged in at [time]." You can look up the *ctime* library and check which of the function is suitable for this.
5. Implement an exit function call **exitProgram()** that exits the program gracefully by displaying a "Goodbye [name]" message. Remember to use the return value from the **getUserName()** function to display the name of the user. After this, call the **exit(0)** function to exit the program gracefully.

## Step 2 Grading system:

Obtain these functions from your previous lab work 06.

6. Create a function called **getNumberOfStudents()** that asks the user to input the number of students in the class. Ensure that the input is a valid number between 1 and 100. Return the number of students as an integer.
7. Create a function called **getStudentData()** that takes the student number (e.g., 1st, 2nd, 3rd) as an argument and allows the user to input the student's name and score. Ensure that the score is a valid number between 0 and 100. Update the relevant statistics (total score, highest score, lowest score, etc.) within this function. You may want to make use of pass by reference for this function as given below.

```
void getStudentData(double &totalScore, double
    &highestScore, double &lowestScore, string
    &studentWithHighestScore, string
    &studentWithLowestScore, int &passCount, int
    &failCount)
```

8. Create a function called **displayStatistics()** that calculates and displays the following statistics:
  - Average score for the class.
  - Highest score and the student's name who achieved it.
  - Lowest score and the student's name who scored it.
  - Number of students who passed (scored 60 or higher).
  - Number of students who failed (scored below 60).

## Step 3:

Main Program Flow:

1. In the main function, call the following functions accordingly to maintain a good program flow.
  - a. Call the **getUserName** function to obtain the name of the user.
  - b. Call the **getUserID** function to obtain the user's ID.
  - c. Call the **calculateASCIISum** function to calculate the ASCII sum of the first 3 characters of the ID.
  - d. Call the **greetUser** function with the results of the functions in parts 3 and 1 as arguments.
2. If the user is a teacher (i.e sum of the first 3 character from **calculateASCIISum** >150), present a menu option using a switch-case statement such that the teacher can navigate the menu for different task. It is defined as follows
  - a. Menu 1: Calculate the student statistics of the students by calling these two functions.
    - i. Call **getNumberOfStudents()** to obtain the number of student
    - ii. Call **getStudentData** to obtain the statistics
  - b. Menu 2: Display the statistics by calling the **displayStatistics** function
  - c. Menu 3: Call the **exitProgram** function to display a goodbye message and exit the program.

3. If the user is a student (i.e sum of the first 3 character from **calculateASCIISum** < 150), present a menu option using a switch-case statement such that the student can navigate the menu for different task. It is defined as follows
  - a. Menu 1: output to the user a “Under Construction” message
  - b. Menu 2: Call the **exitProgram** function to display a goodbye message and exit the program.

#### Step 4:

Test your program by entering sample data for a class of students and verify that it correctly calculates and displays the statistics just as in Lab04.

#### General Hints:

1. Remember to initialize your variables appropriately and at the right location.
2. Implement input validation to ensure user input is within valid ranges.
3. Keep the code organized and use clear variable names.
4. Make sure the program gracefully exits when the user chooses to do so.
5. Feel free to copy the codes from your previous lab.
6. We are assuming the user will only enter the first name as input into the **getUserName** function such that we can use *cin* to collect user’s input

#### Example output 1 (Error)

```
Enter your name: Mi
Enter your 5-character ID: 12fw34
Enter your 5-character ID: 123e4
Welcome teacher Mi
You logged in at Tue Oct 10 05:35:00 2023

Teacher Menu:
1. Calculate student statistics
2. Display statistics
3. Re-run
4. Exit
Enter your choice: 1
Enter the number of students (1 - 100): 353
Enter the number of students (1 - 100): 3
Enter the name of student 1: Wa
Enter the score for Wa: 624
Enter the score for Wa: 23
Enter the name of student 2: we
Enter the score for we: 87
Enter the name of student 3: ty
Enter the score for ty: 12
```

```
Teacher Menu:
1. Calculate student statistics
2. Display statistics
3. Exit
Enter your choice: 2
```

```
Class Statistics:
Average Score: 40.6667
Highest Score: 87 (Student: we)
Lowest Score: 12 (Student: ty)
Number of Students Passed: 1
Number of Students Failed: 2
```

```
Teacher Menu:
1. Calculate student statistics
2. Display statistics
3. Exit
Enter your choice: 4
Goodbye Mi
```

### **Example output 2: Student Menu**

```
Enter your name: key
Enter your 5-character ID: !#)wg
Welcome student key
You logged in at Tue Oct 10 05:41:12 2023
```

```
Student Menu:
1. Under Construction
2. Exit
Enter your choice: 1
This feature is under construction.
```

```
Student Menu:
1. Under Construction
2. Exit
Enter your choice: 2
Goodbye key
```

### **Example output 2: Teacher Menu**

```
Enter your name: Alicia
Enter your 5-character ID: 123er
Welcome teacher Alicia
You logged in at Tue Oct 10 05:48:15 2023
```

```
Teacher Menu:
1. Calculate student statistics
2. Display statistics
3. Exit
Enter your choice: 1
Enter the number of students (1 - 100): 2
Enter the name of student 1: Qw
Enter the score for Qw: 12
Enter the name of student 2: er
Enter the score for er: 76
```

```
Teacher Menu:
```

1. Calculate student statistics
2. Display statistics
3. Exit

Enter your choice: 2

Class Statistics:

Average Score: 44

Highest Score: 76 (Student: er)

Lowest Score: 12 (Student: Qw)

Number of Students Passed: 1

Number of Students Failed: 1

Teacher Menu:

1. Calculate student statistics
2. Display statistics
3. Exit

Enter your choice: 3

Goodbye Alicia

**BONUS (15 points):**

1. Implement a third menu for a student user that allows the students to make simple calculations based on question 2 in lab06.
2. Combine everything in the teacher menu into one such that the conditional statement for teacher only calls the function **teacherMenu()**. This will enhance your knowledge regarding nested functions.
3. Make use of the **getUserName()** function in the **getStudentData** function such that it is called whenever the teacher wants to enter the name of the students.

**Note:** Little/No help will be provided for this bonus. Students are expected to research about it and learn how it is being used.

### 3. How to get full marks

To get a 100% on this lab your code should:

1. Use good variable names such that one can easily understand a variable's purpose just by looking at the name.
2. The program needs to be intuitive (e.g., display proper messages while you are taking user input or printing the result)
3. Follow all good coding conventions such as proper indentation.
4. Adhere to all coding standards outlined in lab2.
5. Follow the instructions of cloning, making dir, and submitting your code to git as previously discussed in lab01 and lab02
6. Comment your code properly (do not write comments for things that are obvious)

7. Push your most recent code in git and submit through canvas as well. The canvas submission should include following files:
  - a. **The cpp file downloaded from git.**
  - b. **At least 2 Image files**
    - i. **Screenshot of the right result when a teacher logs in**
    - ii. **Screenshot of the right result when a student logs in**