

# Lab ACN : C++ Programming Exercises

---

## Exercise 1

Write a temperature conversion program that gives the user the option of converting a value of temperature to Celsius and Fahrenheit scales of temperature.

**Output:**

```
Press 1 to convert Fahrenheit to Celsius
Press 2 to convert Celsius to Fahrenheit

Enter temperature in Fahrenheit: 70
In Celsius that's: 21.111111
```

## Exercise 2 (Automatic Teller Machine)

Write a program that takes balance of a user's account as input. It should then ask the user how much amount he wants to withdraw from his account. The program should take this amount as input and deduct from the balance. Similarly it should ask the user how much amount he wants to deposit in his account. It should take this amount as input and add to the balance. The program shall display the new balance after amount has been withdrawn and deposited.

**Output:**

```
Welcome!
Enter your balance: 1234
-----
Select an option:
1. Withdraw Amount
2. Deposit Amount
3. Exit
-----
How much amount you want to withdraw from your account?
Amount: 1233
Your new balance is: 101
How much amount you want to deposit in your account?
Amount: 100
Your new balance is: 101
Thank you for using the system!
```

Note: Your program should have a check on balance and amount being withdrawn. Amounts greater than balance cannot be withdrawn i.e. balance cannot be negative.

### Exercise 3

Write a program which takes two numbers as input from user and determines which is the larger of the two numbers. The Program should also tell which of the entered numbers is even or odd.

#### Output

```
Welcome
Enter number 1: 4
Enter number 2: 7
number 2 is greater than number 1
number 1 is EVEN.
number 2 is ODD.
```

### Exercise 4

You need to design a calculator that performs four basic arithmetic operations. You will write a program that takes two numbers as input from the user. It then displays a list of arithmetic operations. The user selects one operation and the program displays the result of the corresponding operation. The program should also display which of the two input numbers is greater and which is smaller. Use **do-while** loop. Interaction with the program might look like the following:

```
Please enter two numbers
24
8
Please make a selection:
1) Addition
2) Subtraction
3) Multiplication
4) Division
1
Addition:32
24 is greater.
```

### Exercise 5

Write a program that takes marks of 10 students as input. It calculates the class average and displays it on the screen. Interaction with the program might look like this. Use **while** loop.

```
Welcome
Enter marks: 10
Enter marks: 20
Enter marks: 22.5
Enter marks: 70
Enter marks: 54.75
Enter marks: 87.5
Enter marks: 90.5
Enter marks: 43
Enter marks: 24
Enter marks: 45
Class Average is: 46.725
```

### Exercise 6

Write a program that repeatedly asks the user to enter two money amounts expressed in pounds. The program should then add the two amounts and display the answer in pounds. The program should then convert the amount in Rupees. (Set conversion rate: 1£=120.33 rupees). Display the answer in Rupees. Use a **do-while** loop that asks the user whether the program should be terminated.

```
Welcome!
Enter first amount: £5.10
Enter second amount: £3.2
Total amount: £8.30
Amount in rupees: Rs.998.739/-
Do you wish to continue (y or Y/n or N)?
Thank you for using the system!
```

### Exercise 7

An aircraft system is designed which checks speeds and monitors various functions of the aircraft during the flight. Warnings and errors are generated if any fault develops during the flight. The system is coded in a manner that if 100 is entered as input it indicates “normal speed range”, if 101 is entered it gives “over-speed warning” indicating that aircraft is over-speeding. If 88 is entered, it indicates “Low Oil Pressure” and if 187 is entered it indicates “Engine Failure”. Use **switch statement** to implement this logic.

### Exercise 8

Write a program that calculates and prints the sum of the even integers from 2 to user\_defined\_limit and product of odd integers from 1 to user\_defined\_limit. Use a **do-while** loop that asks the user whether the program should be terminated or not. Also maintain a count as to how many times the user ran the do-while loop.

### Exercise 9 (for loop)

Write the following program and execute it. What output does the program display? Expand the shorthand form of expression `sum += i--` to show what it actually represents. State the total number of iterations made by the “for loop”?

```
int count = 10;
int sum = 0;
for(int i=count; i >= 0; sum += i--)
    ;
cout << sum;
```

### **Exercise 10**

The factorial of an integer  $n$ , written  $n!$ , is simply the product of all the integers from 1 to  $n$ . For example  $5! = 1*2*3*4*5 = 120$ . Write a program to calculate the factorial of integers from 1 to  $n$  where  $n$  is taken from the user. Your output should look like the sample output given below. Use `setw()` operator to format your output. What is the maximum value of factorial that you can calculate for `int` data type? Is there any difference between the maximum value of factorial for unsigned and signed `int`?

```
Enter the value of n: 5

integer      factorial
  1           1
  2           2
  3           6
  4          24
  5         120
```

### **Exercise 11**

Write a program which asks the user whether he wants to draw a pattern as shown in the screenshot below. If user enters “Y” or “y” the program should draw the pattern. If “Q” or “q” is opted the program should quit. However, if any other **alphanumeric input** is given to program the program shall display that the particular input is not recognized. Use nested for loops in your program. Interaction with the program might look like this:

```
Do you want to draw the pattern or quit?
Y (or y) - yes
Q (or q) - Quit
y
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Press any key to continue
```

```
Do you want to draw the pattern or quit?
Y (or y) - yes
Q (or q) - Quit
b
Input not recognized. Program will now exit
Press any key to continue
```

### Exercise 12

A calculator for a supercomputer is to be designed using C language. While the engineers were working on the project they found that the calculator was not giving correct answers for very large calculations. For example, if 150000000 is to be multiplied by 10; the calculator gave faulty result. The bug was detected to be the limited range of integers. Use type casting to design a calculator that performs arithmetic operations of large inputs. (e.g; 150000000 \*10) **Take input from the user in integer data type.**

### Exercise 13

Write a program which takes PIN number of the user as input and then verifies his pin. If pin is verified the program shall display “pin verified” and “Welcome”; otherwise the program shall give user another chance. After 4 wrong attempts, the program shall display “Limit expired” and then exit. Use for loops to implement the logic.

Note: 5 random PIN numbers can be assumed and fed into the program with which input pin is matched.

### Exercise 14

The function definition of starline() function is given to you below. Write a C program that uses the following function to generate the output given below.

```
void starline()
{
    for (int j=0; j<45; j++)
        cout<<'*';
    cout<<endl;
}
```

```
*****
Data Type  Range
*****
char       -128 to 127
short     -32,768 to 32,767
int       System Dependant
long      =2147483648 to 2147483647
*****
Press any key to continue
```

### Exercise 15

Write a program which takes two integers from the user using scanf(). Implement a function

**void swapInts(int, int)** that interchanges the two int values passed to it by main() function. Display the interchanged numbers inside the function using printf(). Also display the values of the two integers in main() function. What do you observe from the output?

What happens to the output if you change the function prototype as follows:

```
void swapInts(int &, int &);
```

### **Exercise 16**

Write a program that takes a set of test scores of a student. It then uses two functions calculateSum() and calculateAverage() to calculate the sum and average of the data set.

Your program must use the following functions:

1. A **value-returning** function, **calculateSum()**, to determine the sum of the test scores. Use a loop to read and sum the test scores. The function does not output the sum of the test scores. This task must be done in the main function.
2. A **void** function, **calculateAverage()**, to determine the average of the test scores. The function does not output the average test score. This task must be done in the main function. Hint: Use pass by reference for average variable.

### **Exercise 17**

If P is the population of the first day of the year, B is the birth rate and D is the death rate, the estimated population at the end of the year is given by formula:

$$P_n + (B * P_n) / 100 - (D * P_n) / 100$$

The population growth rate is given by the formula:

$$B - D$$

Write a program that prompts the user to enter the starting population, birth and death rates and n, the number of years. The program should then calculate and print the estimated population after n years. Your program must contain the following functions:

**GrowthRate:** This function takes as its parameters the birth and death rates and returns the population growth rate.

**Estimated Population:** This function takes as its parameters the current population, population growth rate and n, the number of years. It returns the estimated population after n years. Your program should not accept negative birth rate, negative death rate or population less than 2.

- **Exercise 18**

Write a function that displays a message indicating the number of times it gets called: For instance “The function has been called three times”. Write a program that calls this function 10 times from main(). You will need to use a **global variable** to store the count. A global variable is declared outside the main() function.

Note: All variables declared inside a function are **local variables**. Their value is lost once the function ends.

### **Exercise 19 (Converting a Number from Decimal to Binary)**

This exercise designs a program that uses recursion to convert a non-negative integer in decimal format to equivalent binary number. Let  $x$  be an integer. We call the remainder of  $x$  after division by 2 the right most bit of  $x$ . Thus, the right most bit of 33 is 1 because  $33\% \text{ of } 2$  is 1 and right most bit of 28 is 0 because  $28\% \text{ is } 0$ .

Suppose we want to find the binary representation of 35. First we divide 35 by 2. The quotient is 17 and the remainder, right most bit of 35 is 1. Next we divide 17 by 2. The quotient is 8 and remainder, that is right most bit of 17 is 1. Next we divide 8 by 2. The quotient is 4 and the remainder that is the right most bit of 8 is 0. We continue this process until the quotient is 0.

Thus, to convert an integer “**num**” in base 10 into the equivalent binary number we first convert quotient “**num/2**” into an equivalent binary number, and then append the rightmost bit of “**num**” to the binary representation of “**num/2**”.

#### **Recursive Algorithm :**

decToBin is equal to num if num = 0

decToBin is equal to decToBin(num/base, base) followed by  
cout << remainder if num >0

Write a recursive function **void decToBin ()** having two input parameters **num** and **base** and hence calculate the binary number.

### **Exercise 20 (Generating a pattern)**

Write an iterative function **void printPattern(num)** that takes as a parameter a non-negative integer and generates the following pattern of stars to generate the following pat-

tern. Then try and convert it into a recursive function that takes as a parameter a non-negative integer and generates the following pattern of stars. E.g. if the non-negative integer is 5, then following pattern is generated:

```
* * * * *
* * * *
* * *
* *
*
*
* *
* * *
* * * *
* * *
```

---