

CL-217

Programming

Fundamentals Lab # 11

Objectives:

- Practice and understanding on basic c++ programs
- Control Structure(repetition)
- While loop statement
- Counter control loop variable
- Sentinel controlled loop
- Flag controlled loop
- For loop
- Do-while loop
- Break and continue statement
- Pre and post increment/ decrement operators

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. Use proper **font family (Calibri or Times New Roman)** and **font size of title (16 points), heading (14 points), sub heading (12 points) and normal text (10 points).**
2. First think about problem statement and then write/draw your logic on paper.
3. **Microsoft Visual Studio** should be used to make c++ programs. Programs made with any other software would not be accepted.
4. For each task in the manual create a new C++ program with the naming convention as follows:
TASK-NO
5. **Mention what is happening in each line of code using comments.**
6. Write all codes one by one with proper numbering and also paste screen shot of each problem using **snipping tool**(default screen capture software in windows) on **Microsoft word file.**
7. Please submit your file with this naming convention **ROLLNO_SECTION_GROUPNO_LABNO.**
8. *Do not copy from any source otherwise you will be penalized with negative marks.*
9. Submit your lab on **Google Classroom.**

Problem: 1

When you borrow money to buy a house, a car, or for some other purpose, you repay the loan by making periodic payments over a certain period of time. Of course, the lending company will charge interest on the loan. Every periodic payment consists of the interest on the loan and the payment toward the principal amount. To be specific, suppose that you borrow \$1000 at the interest rate of 7.2% per year and the payments are monthly. Suppose that your monthly payment is \$25. Now, the interest is 7.2% per year and the payments are monthly, so the interest rate per month is $7.2/12 = 0.6\%$. The first month's interest on \$1000 is $1000 * 0.006 = 6$. Because the payment is \$25 and interest for the first month is \$6, the payment toward the principal amount is $25 - 6 = 19$. This means after making the first payment, the loan amount is $1000 - 19 = 981$. For the second

payment, the interest is calculated on \$981. So, the interest for the second month is $981 * 0.006 = 5.886$, that is, approximately \$5.89. This implies that the payment toward the principal is $25 - 5.89 = 19.11$ and the remaining balance after the second payment is $981 - 19.11 = 961.89$. This process is repeated until the loan is paid.

Write a program that accepts as input the loan amount, the interest rate per year, and the monthly payment. (Enter the interest rate as a percentage. For example, if the interest rate is 7.2% per year, then enter 7.2.) The program then outputs the number of months it would take to repay the loan. (Note that if the monthly payment is less than the first month's interest, then after each payment, the loan amount will increase. In this case, the program must warn the borrower that the monthly payment is too low, and with this monthly payment, the loan amount could not be repaid.)

Problem: 2 | do-while, switch case

Write a program that presents the user choice of 5 favorite beverages (Coke, Fanta, Sprite, Pepsi, mineral water) then allow the user to choose a beverage by entering a number 1-5. Output which beverage they chose. User can give choices until he/she has money in his/her account and display message "out of money" with remaining balance when user has not enough money to buy a minimum cost drink (You must use switch statement).

Note:

1. Price of each drink is different and greater than zero (You can initialize prices).
2. Input Balance from user.
3. Input balance must be greater than minimum price drink.
4. Each time user enters his/her choice either he/she want to buy drink or not (use bool data type).
5. Display number of drinks with remaining balance if user did not want to buy drink or out of balance.

Problem: 3 | do-while

Write a program that find the Least Common Multiple by getting input of two numbers from user.

Hint: LCM of two integers a and b is the smallest positive integer that is divisible by both a and b.

Problem: 4 | do-while

Write a program that find the Highest Common Factor by getting input of two numbers from user.

For Example

Hint: The largest integer which can perfectly divide two integers is known as HCF of those two numbers.

Problem: 5 | Removing Break and continue

A criticism of the break and continue statements is that each is unstructured. These statements can always be replaced by structured statements. Describe in general how you'd remove any break statement from a loop in a program and replace it with some structured equivalent. **[Hint:** The break statement leaves a loop from within the body of the loop.

Another way to leave is by failing the loop-continuation test. Consider using in the loop-continuation test a second test that indicates "early exit because of a 'break' condition."]

1. Use the technique you developed here to remove the break statement from the program of

```
1 // Fig. 5.13: fig05_13.cpp
2 // break statement exiting a for statement.
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     int count; // control variable also used after loop terminates
9
10    for ( count = 1; count <= 10; ++count ) // loop 10 times
11    {
12        if ( count == 5 )
13            break; // break loop only if count is 5
14
15        cout << count << " ";
16    } // end for
17
18    cout << "\nBroke out of loop at count = " << count << endl;
19 } // end main
```

2. Use the technique you developed here to remove the continue statement from the program of

```
1 // Fig. 5.14: fig05_14.cpp
2 // continue statement terminating an iteration of a for statement.
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     for ( int count = 1; count <= 10; ++count ) // loop 10 times
9     {
10         if ( count == 5 ) // if count is 5,
11             continue; // skip remaining code in loop
12
13         cout << count << " ";
14     } // end for
15
16     cout << "\nUsed continue to skip printing 5" << endl;
17 } // end main
```

Problem: 6 | While, do-while

Consider the following two loops

```
a. i = 11;
   while (i <= 10)
   {
       cout << i << " ";
       i = i + 5;
   }
   cout << endl;

b. i = 11;
   do
   {
       cout << i << " ";
       i = i + 5;
   }
   while (i <= 10);

   cout << endl;
```

Write two different programs that display the outputs of above mention code and also explain difference between them on word file.

Problem: 7 | do-while

Write a program using do-while that display the number between 1 and 100 by getting input from user. If number is less than 1 or greater than 100 than input again from user.

Problem: 8 |

A high school has 1000 students and 1000 lockers, one locker for each student. On the first day of school, the principal plays the following game:

She asks the first student to go and open all the lockers. She then asks the second student to go and close all the even-numbered lockers. The third student is asked to check every third locker. If it is open, the student closes it; if it is closed, the student opens it. The fourth student is asked to check every fourth locker. If it is open, the student closes it; if it is closed, the student opens it. The remaining students continue this game. In general, the **n**th student checks every **n**th locker. If the locker is open, the student closes it; if it is closed, the student opens it. After all the students have taken their turn, some of the lockers are open and some are closed.

Write a program that prompts the user to enter the number of lockers in a school. After the game is over, the program outputs the number of lockers that are opened. Test run your program for the following inputs: 1000, 5000, 10000. Do you see any pattern developing?

(Hint: Consider locker number 100. This locker is visited by student numbers 1, 2, 4, 5, 10, 20, 25, 50, and 100. These are the positive divisors of 100. Similarly, locker number 30 is visited by student numbers 1, 2, 3, 5, 6, 10, 15, and 30. Notice that if the number of positive divisors of a locker number is odd, then at the end of the game, the locker is opened. If the number of positive divisors of a locker number is even, then at the end of the game, the locker is closed.)

Problem: 9 | Pre/Post Increment & Decrement Operators

Run the program below, comment on each line properly with functionality of that command in main(). Attach output of program also.

```
#include <iostream>
using namespace std;

int main()
{
    int x = 1, a;
    a = x++;
    cout << "\na = " << a;
    x++;
    a = x;
    cout << "\na = " << a;
    a = ++x;
    cout << "\na = " << a;
    ++x;
```



```
a = x;  
cout << "\na = " << a;  
system("pause");  
return 0;  
}
```

Submit your lab work on Google Classroom