



Programming Fundamentals

Lab Manual - Week 07



Introduction

Welcome to your favorite programming Lab. In this lab manual, we shall work together to learn and implement new programming concepts.

Skills to be learned:

- Distinguish the requirement between the use of conditional and counter loops.
- Divide complex problems into smaller easily solvable sub-problems.

Let's do some coding.

Skill: Distinguish the requirement between the use of conditional and counter loops.

Introduction

By this week, you have learned how to write a program that contains conditional statements, and functions. In this class, we will learn about another very powerful concept known as Loops.

Loops are used to execute a number of instructions repeatedly until a condition is satisfied. Loops can be categorized into two major categories.

- Conditional Loops
- Counter Loops

Conditional Loops

Conditional loops help to repeat a set of instructions until some condition is true. There are two common places for its use.

- Reading an unknown amount of input from the user
- Validating input.

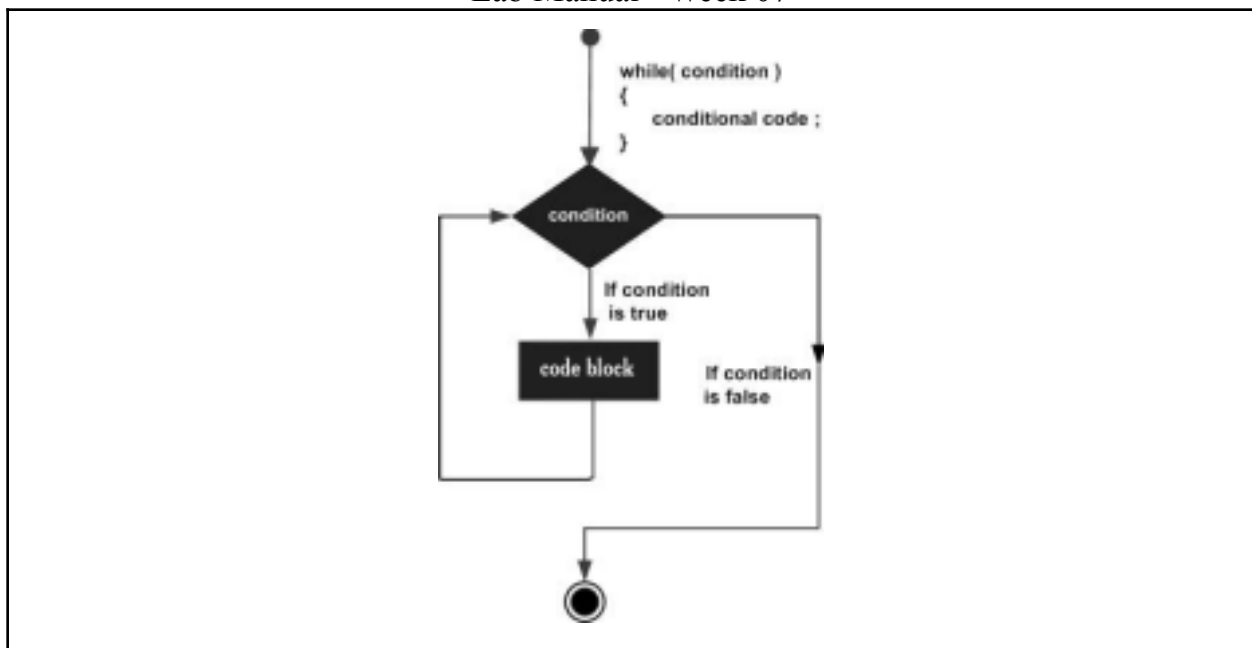
C++ provides a while loop that is used as a conditional loop.

Step 1: Distinguish the requirement between the use of conditional loops and counter loops.



Programming Fundamentals

Lab Manual - Week 07



Task 01(WP): Write a program that keeps printing I am happy on the screen until the user enters **n**.

In this problem, we don't know in advance how many times the loop will be executed.

But we do know the terminating condition. i.e., when the user will enter **n** then we will stop the while loop.

In such situations, we use the While loop which is a conditional loop.

```
#include<iostream>
using namespace std;
main()
{
    char choice = 'y';
    while(choice != 'N' && choice != 'n')
    {
        cout << "I am happy !" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
    }
}
```

Consider the attached solution

Skill Distinguish the requirement between the use of conditional and counter loops.



Programming Fundamentals



Lab Manual - Week 07

Great Work Students, You have just learned another skill. Let's use this skill to solve more complex problems.

Task 02(WP):

Write a Program that keeps asking for inputting a number and adding the sum until the user enters a negative number.

Observe that the program can not move ahead until the user enters a positive number.

```
#include<iostream>
using namespace std;
main()
{
    int num = 0, sum = 0;

    while(num >= 0)
    {
        sum = sum + num;
        cout << "Enter a Number: ";
        cin >> num;
    }

    cout << "Sum: " << sum;
}
```

Consider the attached solution

Counter Loops

The Counter loops are used in situations where the **program knows in advance** how

many times the loop will be executed. The “for” loop is an example of the counter loop.

The **for Loop** Consists of three major components.

- Initial Statement
- Loop Condition
- Update Statement

Look at the following diagram to understand the flow of the For loop. **Skill:** Distinguish

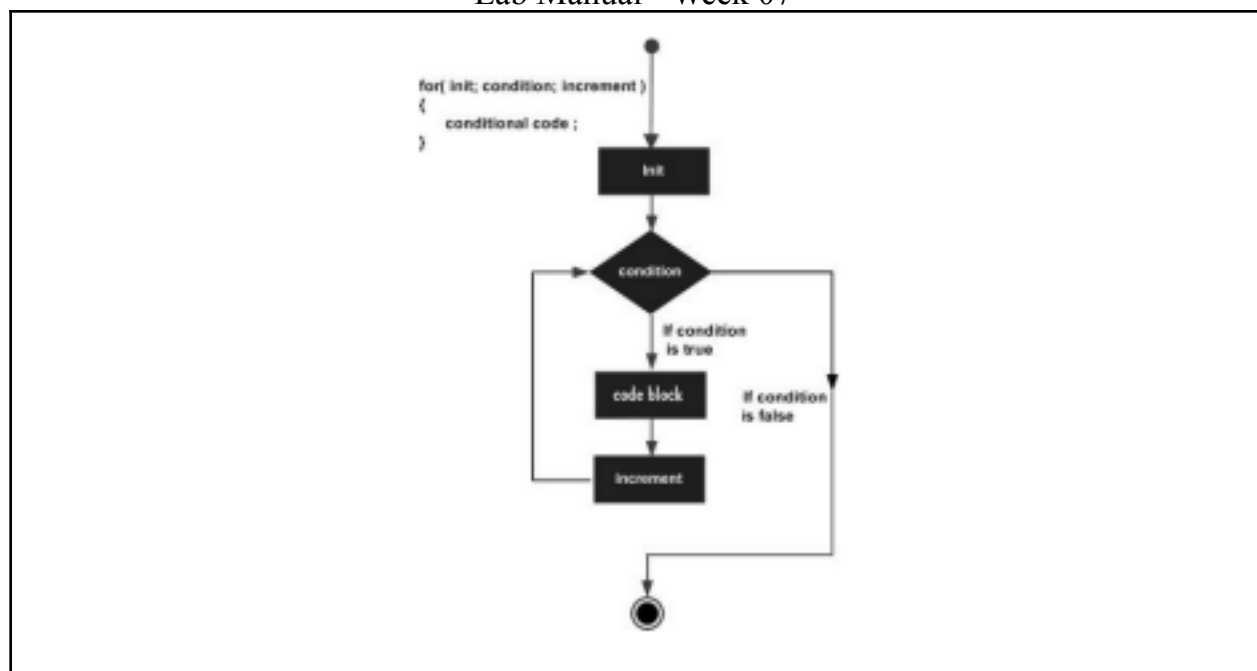
the requirement between the use of conditional and counter loops.



Programming Fundamentals



Lab Manual - Week 07



- The loop **starts** from the **initial statement**
- Now, every time the **condition is True**, the loop **executes the body** and then **executes the update statement**.

This process is repeated until the **condition** is evaluated as **False**.

Consider the following example for better understanding.

Task 03(WP): Write a program that has a function that shows counting from 1 to 10 on the console screen.

Without the loop this program would consist of **10 cout << statements;**

However, as we are performing the same task repeatedly and know in advance that we need to perform the task 10 times.

Therefore, we can use the for loop in this situation.

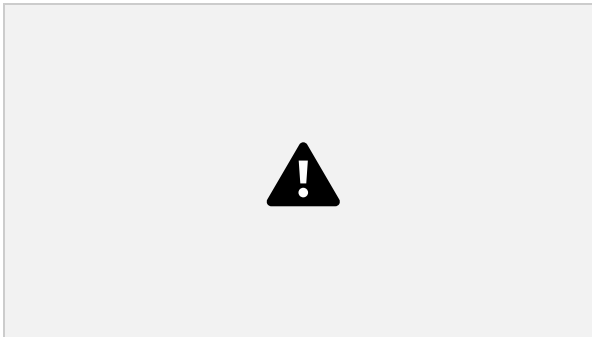


inguish the requirement between the use of condi counter loops.



Programming Fundamentals

Lab Manual - Week 07



Consider the attached solution.

The loop will start from the **initial statement of i=1** and if the **loop condition is true** the **update statement** will increase i by 1 each time the loop body is completed.

Consider the following explanation for better understanding.

Explanation:

		$i \leq 10$	Action (Body of for L	
1st	$i = 1$	True	1 is printed.	i is increased to 2.
2nd	$i = 2$	True	2 is printed.	i is increased to 3.
3rd	$i = 3$	True	3 is printed.	i is increased to 4.
4th	$i = 4$	True	4 is printed.	i is increased to 5.
5th	$i = 5$	True	5 is printed.	i is increased to 6.
6th	$i = 6$	True	6 is printed.	i is increased to 7.
7th	$i = 7$	True	7 is printed.	i is increased to 8.
8 th	$i = 8$	True	8 is printed.	i is increased to 9.

9th $i = 9$ True 9 is printed. i is increased to 10. 10th $i = 10$ True 10 is printed. i is increased to 11. 11th $i = 11$ **false** The loop is terminated

Great Work Students, You have just learned another skill.

Let's use this skill to solve more complex problems.

Task 04(WP): Calculate the sum of the first 5 natural numbers.



inguish the requirement between the use of condi counter loops.



Programming Fundamentals

Lab Manual - Week 07

Consider the above-mentioned question and think about how we can do this by using the loop. **??**

In such problems, we can divide the problem into sub-problems. For example, consider the previous working problem example where we printed the first ten natural numbers on the screen.

Now, what if we can store the number in some variable and add the next number in the previously stored variable after each update statement?

Consider the following solution for better understanding.



Similarly, we can perform similar complex tasks with repetitive structures using the for loop.

Task 05(OP):

Write a program to calculate the sum of the first 100 natural numbers.



inguish the requirement between the use of condi counter loops.



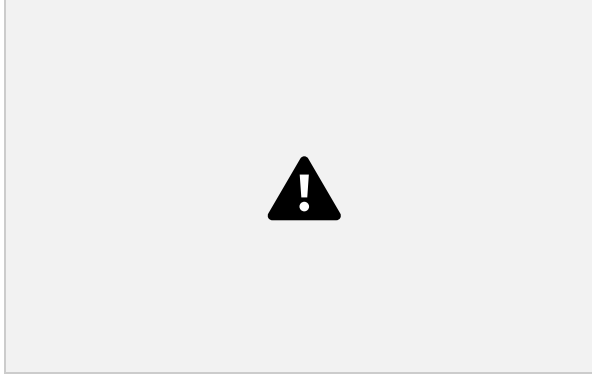
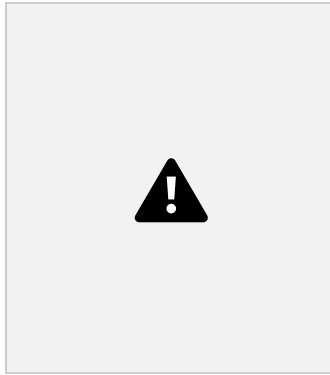
Programming Fundamentals

Lab Manual - Week 07

It's your Choice

Both Loops, may it be a Conditional Loop or a Counter Loop can be used for solving the same problems. However, **it is up to YOU to choose the best suitable option.**

Solution with Conditional Loop Solution with Counter Loop

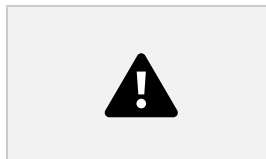


Note: It is better to use **For loop** when we already know how many times the loop has to be executed (We can place the initialization statement, loop condition and increment statement in a single line).

It is better to use the **While loop** when you have to keep repeating something until some condition is met.

Now before moving towards solving the complex real world problems, it's important that we dry run the while and for loops so that we get a deep understanding of how the loops are working.

What will be the Output of following code snippets?








inguish the requirement between the use of condi



counter loops.

Programming Fundamentals

Lab Manual - Week 07

Conclusion

Conditional Loop	We use Conditional Loops in programs where we do not know in advance how many times the loop will be executed. While Loop is an example of a Conditional Loop.
Counter Loops	We use Counter Loops in programs where we know in advance how many times the loop will be executed. For Loop is an example of a Counter Loop.



nguish the requirement between the use of condi counter loops.



Programming Fundamentals

Lab Manual - Week 07

Task 01(CP): (printTable)

Write a function that takes a number from the user and print its multiplication table on the console screen.

```
void printTable(int number);
```

Task 02(CP): (generateFibonacci)

Write a program that prompts the user to input the length of Fibonacci series and then call the function to display the series.



```
void generateFibonacci(int length);
```

Task 03(CP): (totalDigits)

Write a function named “**totalDigits**” that takes the number as input parameter and returns the total number of digits in that number.



Distinguish the requirement between the use of conditional and counter loops.



Programming Fundamentals

Lab Manual - Week 07

Task 04(CP): (frequencyChecker)

Find the frequency of a digit in a number.
Make a function whose prototype will be
int frequencyChecker(int number, int digit);
you have to pass this function a number
and a digit whose frequency you want to
check then the function returns the number
of times the digit occurs in the number.

Test Cases:

frequencyChecker(566960, 6) => 3

frequencyChecker(566960, 5) => 1

Task 05(CP): (digitSum)

Write a function that takes a number from
the user and prints the sum of its digits on
the console screen.

int digitSum(int number);

Task 06(CP): (GCD/LCM)

Write two separate functions to find the
greatest common divisor (GCD) and Least
Common Multiple (LCM) of the given
two numbers.

The greatest **Common Divisor (GCD)** or
Highest Common Factor (HCF) of two
positive integers is the largest positive
integer that divides both numbers without
a remainder.

The **Least Common Multiple (LCM)** of
two integers is the smallest integer that is

a multiple of both numbers. Write the functions with the following prototype.



inguish the requirement between the use of conditional and counter loops.



Programming Fundamentals

Hint:

$$\text{LCM}(a, b) = (a * b) / \text{GCD}(a, b)$$

```
int calculateGCD(int number1, int
                number2);
int calculateLCM(int number1, int
                number2, int gcd);
```

Task 07(CP): (Percentage)

We have n integer numbers within the range of $[1 \dots 1000]$. Some percent of p_1 are under 200, another percent p_2 are from 200 to 399, percent p_3 are from 400 to 599, percent p_4 are from 600 to 799 and the rest p_5 percent are from 800 upwards. Write a program that calculates and prints the percentages p_1 , p_2 , p_3 , p_4 and p_5 .

Example: we have $n = 20$ numbers: 53, 7, 56, 180, 450, 920, 12, 7, 150, 250, 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. We get the following distribution and visualization:

Input Data:

On the first line of the input there is an integer n ($1 \leq n \leq 1000$) that represents the count of lines of numbers that will be passed. On each of the following n lines we have one integer within range of $[1 \dots 1000]$ – numbers, on which we have to calculate the histogram.

Output Data:

Print on the console a histogram that consists of 5 lines, each of them containing a number within the range of $[0\% \dots 100\%]$, formatted up to two digits after the decimal point (for example 25.00%, 66.67%, 57.14%).

Skill: Distinguish the requirement between the use of conditional and counter loops.



Programming Fundamentals



Lab Manual - Week 07



Distinguish the requirement between the use of conditional and counter loops.



Programming Fundamentals



Task 08(CP): (Lilly)

Lilly is N years old. For each birthday she receives a present. For each odd birthday (1, 3, 5, ..., n) she receives toys, and for each even birthday (2, 4, 6, ..., n) she receives money. For her second birthday she received 10.00 USD, and the amount is increased by 10.00 USD for each following even birthday (2 → 10, 4 → 20, 6 → 30 etc.). Over the years Lilly has secretly saved her money. Lilly's brother, in the years when she received money, took 1.00 USD from each of the amounts. Lilly has sold the toys, received over the years, each one for P USD and added the sum to the amount of saved money. With the money she wanted to buy a washing machine for X USD.

Write a **function** that calculates how much money she has saved and if it is enough to buy a washing machine.

```
int calculateMoney(int age, int price, int price);
```

Input Data

We read from the console 3 numbers, each on a separate line:

- Lilly's age – integer in the range of [1 ... 77].
- Price of the washing machine – integer number in the range of [1 ... 10 000].
- Unit price of each toy – integer in the range of [0 ... 40].

Output Data

Print on the console one single line:

- If Lilly's money is enough:
- “Yes! {N}” – where N is the remaining money after the purchase
- If the money is not enough:
- “No! {M}” – where M is the insufficiency amount

Distinguish the requirement between the use of conditions and counter loops.



Programming



Fundamentals

Lab Manual - Week 07

Sample Output:

Input Output	
10	Yes!
170	5
6	
21	No!
1570	1000
3	



Task 09(CP): (Ivan)

Ivan is 18 years old and receives an inheritance that consists of X money and a time machine. He decides to return to 1800, but does not know if the money will be enough to live without working. Write a program that calculates if Ivan will have enough money to not have to work until a particular year (inclusive). Assuming that for every even (1800, 1802, etc.) year he will spend 12000 dollars. For every odd one (1801, 1803, etc.) he will spend $12000 + 50 * [\text{the age he will have reached in the given year}]$.

int calculatePrice(int money, int year);

Input Data

The input is read from the console and contains exactly 2 lines:

- Inherited money – integer number in the range [1 ... 1 000 000].
- Year, until which he has to live in the past (inclusive) – integer number in the range [1801 ...

1900].

Output Data



nguish the requirement between the use of condi counter loops.



Programming Fundamentals

Lab Manual - Week 07

Print on the console 1 line. The sum must be formatted up to the two symbols after the decimal point:

- If money is enough:
 - "Yes! He will live a carefree life and will have {N} dollars left." – where N is the money that will remain.
- If money is NOT enough:
 - "He will need {M} dollars to survive." – where M is the sum that is NOT enough.

Sample Output:

Input Output	
50000	Yes! He will live a carefree life and will have 13050 dollars left.
1802	
100000	He will need 12400 dollars to survive.
1808	



Good Luck and Best Wishes !!

Happy Coding ahead :)

Skill: Distinguish the requirement between the use of conditional and counter loops.