Programming Fundamental

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 counter and conditional loops.
- Divide complex problems into smaller easily solvable sub-problems.
- Differentiate between the continue and break statements.

Let's do some coding.

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

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

- Counter Loops
- Conditional Loops

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.

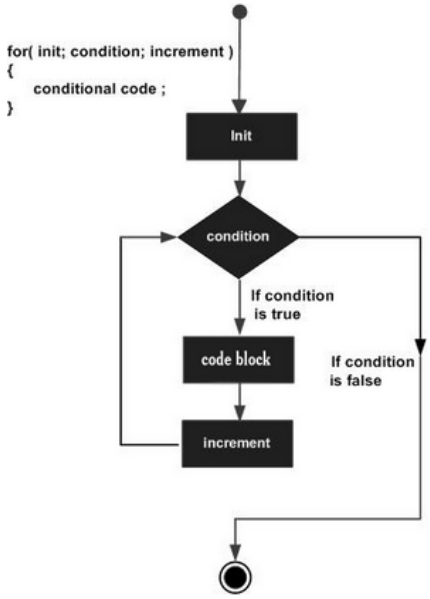
The **for Loop** Consists of three major components.

- Initial Statement
- Loop Condition
- Update Statement

Look at the following diagram to understand the flow the For loop.

Programming Fundamental

Lab Manual - Week 07



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

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

Consider the following example for better understanding.

Task 01(WP): Write a program that has a function that shows counting from 1 to 10 on 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.

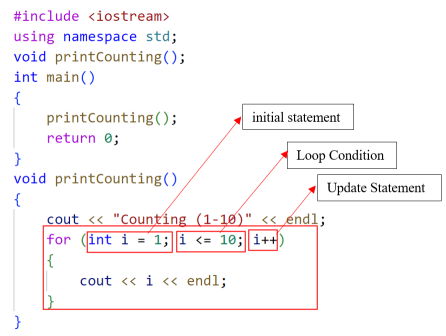
Therefore, we can use the for loop in this situation.

```
D:\PF codes>c++ week07.cpp -o week07.exe

D:\PF codes>week07.exe
Counting (1-10)
1
2
3
4
5
6
7
8
9
10

D:\PF codes>
```

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

Programming Fundamental

Lab Manual - Week 07




```
#include <iostream>
using namespace std;
void printCounting();
int main()
{
    printCounting();
    return 0;
}
void printCounting()
{
    cout << "Counting (1-10)" << endl;
    for (int i = 1; i <= 10; i++)
    {
        cout << i << endl;
    }
}
```

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:

Iteration	Variable	i <= 10	Action	Update Statement
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.
8th	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 02(WP): Calculate the sum of the first 5 natural numbers.

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

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

```
D:\PF codes>c++ example.cpp -o example.exe

D:\PF codes>example.exe
15

D:\PF codes>
```

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.

```
#include <iostream>
using namespace std;
int showSum();
main()
{
  int total;
  total = showSum();
  cout << total << endl;
}
int showSum()
{
  int sum = 0;
  for (int i = 1; i <= 5; i = i + 1)
  {
    sum = sum + i;
  }
  return sum;
}
```

We are storing the result of each iteration in **sum** and updating its value after each iteration.

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

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

Programming Fundamental

Lab Manual - Week 07

Task 03(CP):

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

void printTable(int number);

Task 04(CP):

Write a procedure that prompts the user to input the length of Fibonacci series and display the series.

Fibonacci Number Series

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811...

```
D:\PF codes>c++ fibonacci.cpp -o fibanacci.exe

D:\PF codes>fibanacci.exe
How many numbers of Fibonacci Series you want to print: 4
0, 1, 1, 2
D:\PF codes>fibanacci.exe
How many numbers of Fibonacci Series you want to print: 6
0, 1, 1, 2, 3, 5
D:\PF codes>fibanacci.exe
How many numbers of Fibonacci Series you want to print: 2
0, 1
D:\PF codes>
```

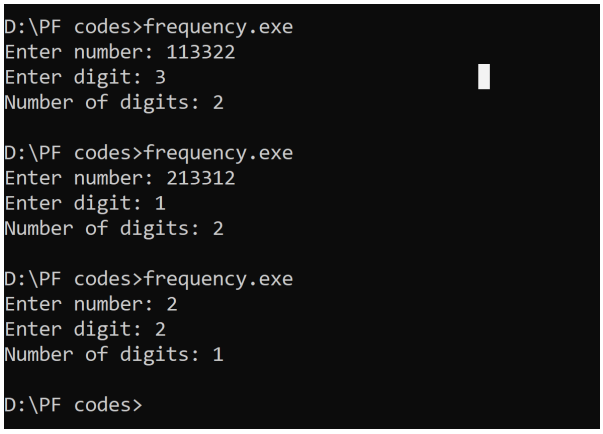
Task 05(CP):

Write a function named “totalDigits” that return the total number of digits in a number.

```
D:\PF codes>c++ totalDigits.cpp -o totalDigits.exe

D:\PF codes>totalDigits.exe
Enter number: 13
Number of digits: 2

D:\PF codes>totalDigits.exe
Enter number: 16
Number of digits: 2

D:\PF codes>totalDigits.exe
Enter number: 78943789
Number of digits: 8

D:\PF codes>
```

Task 06(CP):

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

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 07(CP):

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);**	12345 // 1 + 2 + 3 + 4 + 5 = 15 123 // 1 + 2 + 3 = 6 12 // 1 + 2 = 3

Conditional Loops

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

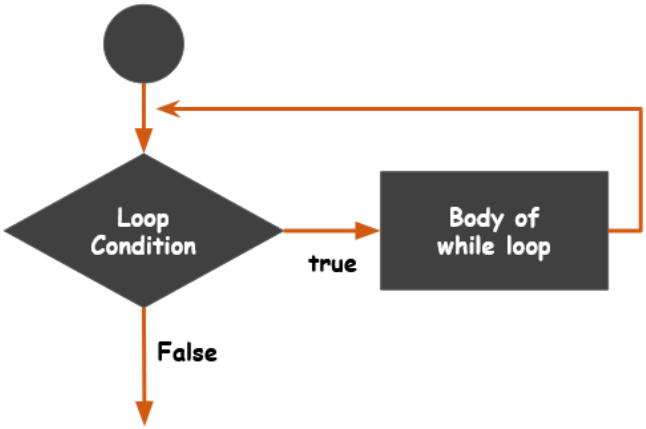
- Reading an unknown amount of input from the user
- Validating input.

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

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

Programming Fundamental

Lab Manual - Week 07



Task 08(WP): Write a program that keeps printing I am happy on the screen until the user enters **y**.

In this problem, we don't know in advance how many times the loop will be executed.
 	
In such situations, we use the While loop which is a conditional loop.
	`D:\PF codes\week05>c++ week07.cpp -o week07.exe`
	
	`D:\PF codes\week05>week07`
	`I am happy !`
	`Enter your choice: y`
	`I am happy !`
	`Enter your choice: y`
	`I am happy !`
	`Enter your choice: y`
	`I am happy !`
	`Enter your choice: y`
	`I am happy !`
	`Enter your choice:`

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

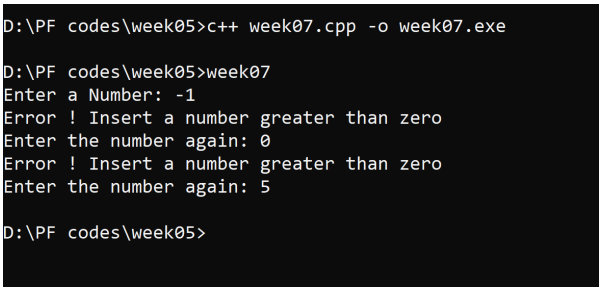
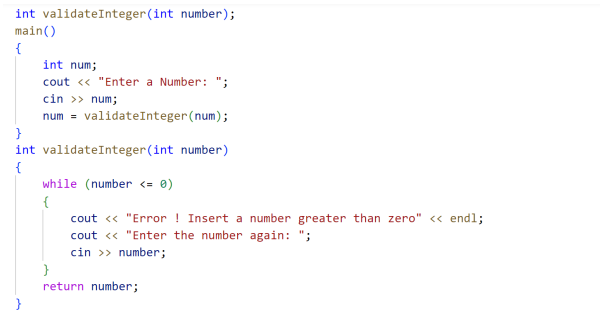
Great Work Students, You have just learned another skill.

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

Task 09(WP):

Write a Program that keeps asking for inputting a number until the user enters a positive number

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

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

Task 10(OP):

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

Its your Choice 💡

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

Consider the following task for better understanding

Task 11(WP):

Write a program that takes a number from the user and prints the counting until that number.

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

```
D:\PF codes\week05>c++ week07.cpp -o week07.exe

D:\PF codes\week05>week07
Enter a Number: 10
** Counting with For Loop **
        1
        2
        3
        4
        5
        6
        7
        8
        9
        10
** Counting with While Loop **
        1
        2
        3
        4
        5
        6
        7
        8
        9
        10

D:\PF codes\week05>
```

We Can use both of the loops to perform certain tasks, however, it is up to the programmer to use the best loop according to the requirement of the program.

```
void forCounting(int number);
void whileCounting(int number);
main()
{
    int num;
    cout << "Enter a Number: ";
    cin >> num;
    forCounting(num);
    whileCounting(num);
}
void forCounting(int number)
{
    cout << "** Counting with For Loop **" << endl;
    for (int i = 1 ; i<= number; i++)
    {
        cout << "\t" << i << endl;
    }
}
void whileCounting(int number)
{
    cout << "** Counting with While Loop **" << endl;
    int i = 1;
    while(i <= number)
    {
        cout << "\t" << i << endl;
        i = i + 1;
    }
}
```

Consider the attached solution for a better understanding

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.
Counting Loops	We use Counting Loops in programs where we know in advance how many times the loop will be executed. **For Loop** is an example of a Counting Loop.

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

Task 12(CP):

Write two separate functions to find the greatest common divisor (GCD) or highest common factor (HCF) 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.

Hint:

LCM(a, b) = (a * b) / GCD(a, b)

Write the functions with the following prototype that prints the percentage.

int calculateGCD(int number1, int number2);

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

Task 13(CP):

We have n integer numbers within the range of [1 … 1000]. Some percent of them p1 are under 200, another percent p2 are from 200 to 399, percent p3 are from 400 to 599, percent p4 are from 600 to 799 and the rest p5 percent are from 800 upwards. Write a program that calculates and prints the percentages p1, p2, p3, p4 and p5.

Write a function with the following prototype that prints the percentage.

void printPercentage(int number);

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 … 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% … 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 counter and conditional loops.

Sample Input and Output

Input	Output	Input	Output
3 1 2 999	66.67% 0.00% 0.00% 0.00% 33.33%	**4** 53 7 56 999	75.00% 0.00% 0.00% 0.00% 25.00%

Input	Output	Input	Output
7 800 801 250 199 399 599 799	14.29% 28.57% 14.29% 14.29% 28.57%	**9** 367 99 200 799 999 333 555 111 9	33.33% 33.33% 11.11% 11.11% 11.11%

Input	Output
14 53 7 56 180 450 920 12 7 150 250 680 2 600 200	57.14% 14.29% 7.14% 14.29% 7.14%

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

Task 14(CP):

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.

float 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 – number in the range of [1.00 … 10 000.00].
- 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
- Numbers N and M must be formatted up to the second digit after the decimal point.

Sample Output:

Input	Output
10 170.0 6	Yes ! 5.00
21 1570.98 3	No ! 997.98

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

Programming Fundamental

Lab Manual - Week 07

Task 15(CP):

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 12 000 dollars. For every odd one (1801, 1803, etc.) he will spend 12 000 + 50 * [the age he will have reached in the given year].

float calculatePrice(float money, int year);

Input Data

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

- Inherited money – a real number in the range [1.00 … 1 000 000.00].
- Year, until which he has to live in the past (inclusive) – integer number in the range [1801 … 1900].

Output Data

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 1802	Yes! He will live a carefree life and will have 13050.00 dollars left.
100000.15 1808	He will need 12399.85 dollars to survive.

Good Luck and Best Wishes !!

Happy Coding ahead :)

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