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. 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.

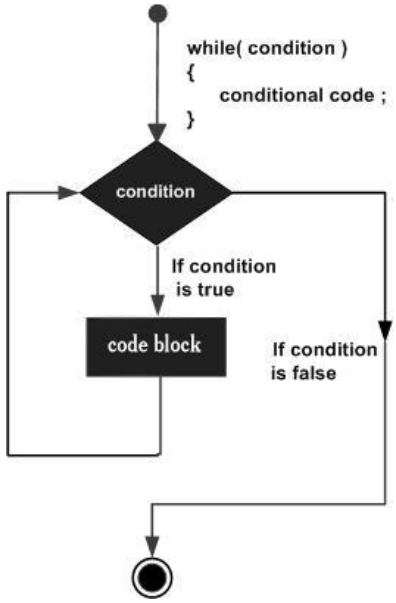
- Reading an unknown amount of input from the user
- Validating input.

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

Skill: Distinguish the requirement between the use of conditional 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.	G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task.exe I am happy ! Enter your choice: y I am happy ! Enter your choice: d I am happy ! Enter your choice: e I am happy ! Enter your choice: N

`#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 inputing a number and adding the sum until the user enters a negative number.

`G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task.exe`
`Enter a Number: 1`
`Enter a Number: 2`
`Enter a Number: 3`
`Enter a Number: 4`
`Enter a Number: -1`
`Sum: 10`
	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

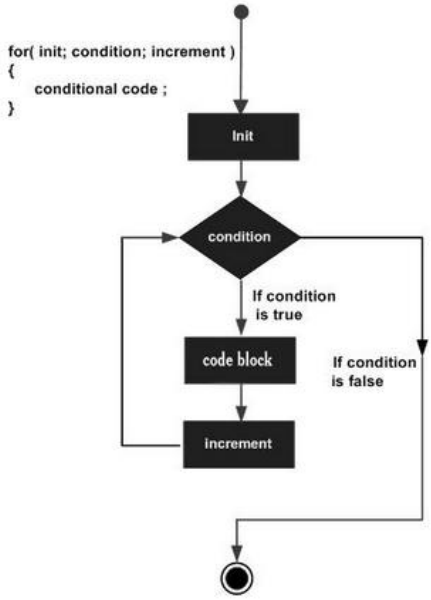
The Counter loops are used in situations where the **program knows in advance** how many times the loop will be executed. The “for” loop is an example of the counter loop.

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.



- 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 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.

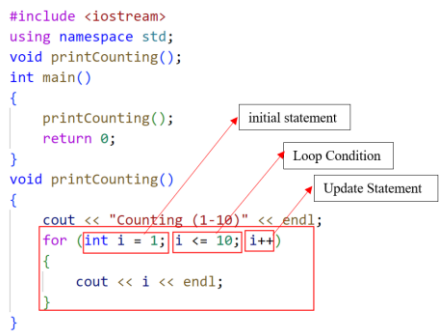
Therefore, we can use the for loop in this situation.

```
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 conditional and 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:

Iteration	**Variable**	**i <= 10**	**Action (Body of for Loop)**	**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 04(WP): Calculate the sum of the first 5 natural numbers.

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

Task 05(OP):

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

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task.exe
Sum: 5050
```

Skill: Distinguish the requirement between the use of conditional and 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

`#include<iostream>`
`using namespace std;`
`main()`
`{`
`  int num = 1, sum = 0;`
 	
`  while(num <= 100)`
`  {`
`  sum = sum + num;`
`  num = num + 1;`
`  }`
`  cout << "Sum: " << sum;`
`}`
	`#include<iostream>`
	`using namespace std;`
	`main()`
	`{`
	`  int sum = 0;`
	
	`  for(int num = 1; num <= 100; num = num + 1)`
	`  {`
	`  sum = sum + num;`
	`  }`
	`  cout << "Sum: " << sum;`
	`}`

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?

```
for (int i = 0; i < 5; i++) {
    if (i % 2 == 0)
        cout << i << " ";
}
```

```
int i = 0;
while (i < 5) {
    cout << i << " ";
    i += 2;
}
```

```
int sum = 0;
for (int i = 1; i <= 5; i++) {
    sum += i;
}
cout << sum;
```

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

Programming Fundamentals

Lab Manual - Week 07

```
int i = 5;
while (i > 0) {
    cout << i << " ";
    i--;
}
```

```
int product = 1;
for (int i = 1; i <= 5; i++) {
    product *= i;
}
cout << product;
```

```
int i = 0;
while (i < 10) {
    cout << i << " ";
    i += 3;
}
```

```
for (int i = 5; i > 0; i--) {
    cout << i << " ";
}
```

```
int i = 0;
while (i < 5) {
    cout << i << " ";
    i++;
}
```

```
int sum = 0;
for (int i = 10; i > 0; i -= 2) {
    sum += i;
}
cout << sum;
```

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.

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

Programming Fundamentals

Lab Manual - Week 07

Task 01(CP): (printTable)

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

void printTable(int number);

```
G:\Semesters\Programming Fundamentals
Enter a number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Task 02(CP): (generateFibonacci)

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

void generateFibonacci(int length);

```
G:\Semesters\Programming Fundamentals (Fall
Enter the length of the Fibonacci series: 6
0, 1, 1, 2, 3, 5

G:\Semesters\Programming Fundamentals (Fall
Enter the length of the Fibonacci series: 2
0, 1

G:\Semesters\Programming Fundamentals (Fall
Enter the length of the Fibonacci series: 1
0
```

Task 03(CP): (totalDigits)

Write a code named **"totalDigits"** that takes the number as input parameter and returns the total number of digits in that number.

```
G:\Semesters\Programming Fundamentals
Enter a number: 0
Total number of digits: 1

G:\Semesters\Programming Fundamentals
Enter a number: 345
Total number of digits: 3

G:\Semesters\Programming Fundamentals
Enter a number: 123456789
Total number of digits: 9

G:\Semesters\Programming Fundamentals
Enter a number: -4321
Total number of digits: 4
```

Skill: 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. You have to take number as input for the user and number whose frequency you want to calculate. **Test Cases:** frequencyChecker(566960, 6) => 3 frequencyChecker(566960, 5) => 1	G:\Semesters\Programming Fundamentals Enter a number: 1122 Enter the digit to check: 1 Frequency: 2 G:\Semesters\Programming Fundamentals Enter a number: 2200000 Enter the digit to check: 0 Frequency: 5

Task 05(CP): (digitSum)

Write a code that takes a number from the user and prints the sum of its digits on the console screen. (In this number could be of any length last time the number was 4 digit number fixed) **int digitSum(int number);**	G:\Semesters\Programming Fundamentals Enter a number: 222 Sum of digits: 6 G:\Semesters\Programming Fundamentals Enter a number: 4321 Sum of digits: 10

Task 06(CP): (GCD/LCM)

Write a program to calculate the GCD and LCM of a number. 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)**	G:\Semesters\Programming Fundamentals Enter the first number: 23 Enter the second number: 32 GCD: 1 LCM: 736 G:\Semesters\Programming Fundamentals Enter the first number: 44 Enter the second number: 66 GCD: 22 LCM: 132 G:\Semesters\Programming Fundamentals Enter the first number: 17 Enter the second number: 23 GCD: 1 LCM: 391

Task 07(CP): (Percentage)

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

Programming Fundamentals

Lab Manual - Week 07

We have n integer numbers within the range of [1 … 1000]. Some percent of 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.

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 ≤ n ≤ 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%).

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%

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

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%

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task7.exe
Enter numbers count: 7
Enter a number: 800
Enter a number: 801
Enter a number: 250
Enter a number: 199
Enter a number: 399
Enter a number: 599
Enter a number: 799
14.29%
28.57%
14.29%
14.29%
28.57%
```

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

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

one for P USD and added the sum to the amount of money saved. With the money she wanted to buy a washing machine for X USD.

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

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

Sample Output:

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

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task8.exe
Enter Lilly's age: 21
Enter the price of the washing machine: 1570
Enter the unit price of each toy: 3
No!
1000
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task8.exe
Enter Lilly's age: 10
Enter the price of the washing machine: 170
Enter the unit price of each toy: 6
Yes!
5
```

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

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 * [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

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 dollars left.
100000 1808	He will need 12400 dollars to survive.

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

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task9.exe
Enter Money: 50000
Enter Year: 1802
Yes! He will live a carefree life and will have 13050 dollars left.

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 7\Lab Tasks>Task9.exe
Enter Money: 100000
Enter Year: 1808
He will need 12400 dollars to survive.
```

Good Luck and Best Wishes !!

Happy Coding ahead :)

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