**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE110** | **Credits: 1.5** |
| **Course Name: Programming Language 1** | **Semester: FALL’18** |

**Lab 04  
Introduction to the Concept of Loops and Flowchart using Loops**

1. **Topic Overview:**

The students will solve problems to familiarize themselves with loops and necessity of their use. In their previous labs, they did not use any loops. They will also learn how a loop decreases the redundancy of a program through these tasks. There are in total of 11 problems in this lab.

1. **Lesson Fit:**

Pre-requisite to this lab is Lab1 to Lab3.

1. **Learning Outcome:**

After this lecture, the students will be able to:

* 1. Print sequences of values using loops
  2. Use nested if/else
  3. Use nested loops

1. **Anticipated Challenges and Possible Solutions**
   1. Task 1 –Task11: Students will make mistakes with the direction of loops and use of connectors
   2. Task 7 : Students use single condition with 63 for implying multiples of both 7 and 9

**Solutions:**

* + 1. Use two separate conditions for detecting multiples of 7 and 9.
  1. Task2- Task11: Students will use logical operators && and ||.   
     **Solutions:**
     1. Make them use separate conditions which will gradually help them to understand conditions (if-else) better.

1. **Acceptance and Evaluation**

Students will show their progress as they complete each problem. They will be marked according to their class performance. Their maybe students who might not be able to finish all 11 tasks, they will submit them later and give a viva to get their performance mark.

1. **Activity Detail**
   1. **Hour: 1  
      Discussion:**Explain how loops works and why they are used. How the loops are represented in flowcharts.  **Problem Task:**
      1. Task 1 (a to d) (Page 3)
   2. **Hour: 2**

**Discussion:**

Review the nested if-else conditions of Lab3 (previous lab), then solve these related problems using loops. Check Task 1 while the students continue with the rest.

**Problem Task:**

* + 1. Task 2 to 6 (Page 3 )
  1. **Hour: 3**

**Discussion:**

Check Task 2 to 6 while the students continue with the rest.

**Problem Task:**

* + 1. Task 7 to 10(Page 3 to Page 4)

1. **Home tasks**
   1. Task 11 (Page-4)
   2. Unfinished tasks

**Lab 4 Activity List**

**Task 1**

Draw four separate flowchart of a program, which prints the following sequences of values in loops:

* 1. 24, 18, 12, 6, 0, -6
  2. -10, -5, 0, 5, 10, 15, 20
  3. 18, 27, 36, 45, 54, 63
  4. 18,-27,36,-45,54,-63

**Task 2**

Draw flowchart of a program, which adds all numbers that are multiples of both 7 and 9 up to 600.

**Task 3**

Draw flowchart of a program, which adds all numbers that are multiples of either 7 or 9 or both up to 600. Ensure that numbers like 63 are added only once in the sum.

**Task 4**

Draw flowchart of a program, which adds all numbers that are multiples of either 7 or 9 but not both, up to 600.

**Task 5**

Draw the flowchart of a program that asks the user for 20 numbers and prints if those numbers are odd or even.

**Task 6**

Draw the flowchart of a program that asks the user for a *quantity*, then takes that many numbers and prints if those numbers are odd or even.

**Task 7**

Draw the flowchart of a program that asks the user for a *quantity* then takes that many numbers and prints the maximum, minimum and average of those numbers.

**Task 8**

Draw the flowchart of a program that takes a number from user and prints the divisors of that number and then how many divisors there were.   
If user gives 6, your program should print: 1, 2, 3, 6. Total 4 divisors.  
If user gives 121, your program should print 1, 11,121.Total 3 divisors.

**Task 9**

An integer number is said to be a perfect number if its factors, including 1 but not the number itself, sum to the number. For example, 6 is a perfect number because factors of 6 are 1, 2, 3, 6 and if we add all factors except itself, 6 = 1 + 2 + 3.   
Draw flowchart of a program, which takes a number and tells if it is a perfect number or not.   
  
Hint: Modify Task-8.

**Task 10**

Draw flowchart of a program that asks user for one number and tells if it is a prime number or not.   
  
Hint: use the divisor count from task 8. If a number has only two divisors, (1 and itself), then it is prime. If it is divisible by more numbers, then it is not a prime.

**Task 11**

Draw flowchart of a program that prints all prime numbers between 40 and 50.

Hint: Modify Task-10.