
CPT 111 – PRINCIPLES OF PROGRAMMING
WEEK 5: PROGRAMMING LAB
Repetition / Looping Control Structure

For every questions, do all type of loops covered in the syllabus: `while` loop, `for` loop and `do-while` loop. Your TA however can only cover **one type of loop** for each question within two hours (if possible).

1. Sum of Numbers

Write a program that asks the user for a positive integer value. The program should use a loop to get the sum of all the integers from 1 up to the number entered. For example, if the user enters 50, the loop will find the sum of 1, 2, 3, 4,... 50.

Input Validation: Do not accept a negative starting number.

2. Characters for the ASCII Codes

Write a program that uses a loop to display the characters for the ASCII codes 0 through 127. Display 16 characters on each line.

3. Multiplication Tables

Write a program that let's the user enter a number and then displays the multiplication table till 10 for that number

4. Conversion Table from Degrees to Radians

Write a program to generate a table of conversion from degrees to radians. The value of degrees should start with 0° and end with 90°. Increment each row from the previous one by only 10°. (Note that $\pi = 180^\circ = 3.141593$ radians.)

5. Calories Burned

Running on a particular treadmill, you burn 3.6 calories per minute. Write a program that uses loop to display the number of calories burned after 5, 10, 15, 20, 25 and 30 minutes.

6. Distance Traveled

The distance a vehicle travels can be calculated as follows:

$\text{distance} = \text{speed} * \text{time}$

For example, if a train travels 40 miles per hour for 3 hours, the distance traveled is 120 miles.

Write a program that asks the user for the speed of a vehicle (in miles per hour) and how many hours it has traveled. The program should then use a loop to display the distance the vehicle has traveled for each hour of that time. Here is an example of the output:

What is the speed of the vehicle in mph? 40
How many hours has it traveled? 3
Hour Distance Traveled

1 40
2 80
3 120

Input Validation: Do not accept a negative number for speed and do not accept any value less than 1 for time traveled.

7. Arithmetic Series

Write a program that asks the user for three values: an initial value for an arithmetic series, a second value for the series, and the number of values that should be calculated and displayed. The difference between the second and the first value determines the increment for the series.

Input Validation: The number of values has to be greater than 0.

Alternate Version: Write a program that takes as the third value an upper limit for the series. If the user enters 100 as the third value, the series stop as soon as it equals or exceed 100.

8. Palindrome Check

A number is called a palindrome if it remains unchanged when its digits are reversed. Write a program to accept a nonzero positive number. The program should loop until the user inputs a valid number. The program should then generate the reversed number. If the reversed number is equal to the original, a message should be displayed that the number is a palindrome.

Hint: To find the reverse of a number, you can follow this procedure to reverse a number n . Initialise reversed number, $revno$ to 0. Divide n by 10 and store the remainder in r . Then add the remainder r to $revno$ multiplied by 10. Repeat continually until n becomes 0.

9. Pattern Generator

Write a program that lets the user input in the following order:

- A character
- An integer 'n' for the number of times this character has to be repeated
- Another character
- Another integer 'm' for the number of times the second character has to be repeated
- Two integers for the number of rows and columns

The program should generate a pattern based on the input parameters. The pattern is composed of the repetition of 'n' times the first character and 'm' times the

second character up until all rows have been filled with the specified number of characters.

If, for example, '*' must be repeated three times and '+' twice in 4 rows of 21 column, the pattern looks like this:

```
***+***+***+***+***+
**++**++**++**++**++
*++***++***++***++***
++***++***++***++***+
```

10. Random Number Guessing Game

Write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." The program should use a loop that repeats until the user correctly guesses the random number.

Alternatively: Enhance the program so it keeps a count of the number of guesses that the user makes. When the user correctly guesses the random number, the program should display the number of guesses.