
Lab Assignment 4

Linear Loops



CSE110: Programming Language I

No of Tasks			Points to Score
<i>Classwork</i>	Evaluation	Homework	Homework $6 \times 10 = 60$
3	2	6	Assessment $2 \times 10 = 20$

The students must complete the classwork tasks in the lab class to obtain the lab performance marks. They will also be marked based on the assessment tasks. The lab instructors may show/explain a few of the classwork tasks to the students if necessary. Any plagiarism in classwork or homework will lead to the student getting zero in the entire assignment. A random viva may take place.

Classwork

1. Write a Java code that would print the following sequences **using while loop**:

a) 24, 18, 12, 6, 0, -6

b) -10, -5, 0, 5, 10, 15, 20

2. Write a Java program that will take N numbers from the user and find their sum and average **using a for loop**.

Sample Input:

N = 5

Input the 5 numbers:

1

2

3

4

5

Expected Output:

The sum of 5 no is: 15

The Average is: 3.0

3. Write a Java program that will keep taking integer numbers as inputs from the user and print the square of those numbers until it gets a negative number and then stop.

Sample Input/Output: (The purple numbers are input.)

Enter Number: 2

$2^2 = 4$

Enter Number: 6

$6^2 = 36$

Enter Number: 1

$1^2 = 1$

Enter Number: 4

$4^2 = 16$

Enter Number: -5

Evaluation

1. Write a Java program that will take an integer as input and print all the divisors of that number.

Sample Input

6

Sample Output

Divisors of 6:

1

2

3

6

2. Write a Java code that asks an integer as input from the user and takes that many integer inputs. Your task is to count how many numbers are non-negative and negative.

Sample Input: (The purple numbers are input.)

Enter an integer: 9

Enter number 1: -8

Enter number 2: 33

Enter number 3: -100

Enter number 4: 10

Enter number 5: 0

Enter number 6: 5

Enter number 7: 10

Enter number 8: -4

Enter number 9: 4

Sample Output:

6 Non-negative Numbers

3 Negative Numbers

Homework

1. Draw a Flowchart and write a Java program that displays the sum of n odd natural numbers.

Sample Input:

Input number of terms: 5

Expected Output:

The odd numbers are:

1

3

5

7

9

The Sum of odd Natural Numbers up to 5 terms is: 25

2. Write a Java program that will read 10 numbers from the user, and then print the first number, the sum of the first 2 numbers, the first 3 numbers, and so on up to the sum of 10 numbers.

Sample Input/Output: (The purple numbers are input.)

Enter Number: 6

Sum = 6

Enter Number: 7

Sum = 13

Enter Number: 2

Sum = 15

Enter Number: 8

Sum = 23

Enter Number: 7

Sum = 30

Enter Number: 5

Sum = 35

Enter Number: 1

Sum = 36

Enter Number: -12

Sum = 24

Enter Number: 0

Sum = 24

Enter Number: 1

Sum = 25

3. Draw a flowchart and write a Java program that will take a positive integer n as input and print all the numbers between 0 to n which are **divisible by 5 but not divisible by 3**.

Sample Input

40

Sample Output

5
10
20
25
35
40

4. Write a program in Java that asks the user for an integer input and counts the number of digits in the number.

Hint: You may keep dividing the number by ten and count how many times this can be done until the number becomes 0.

Sample Input:

7546

Sample Output:

Total digits = 4

5. Write a program in Java that asks the user for an integer input, and print the individual digits forward (From left to right).

Sample Input

32768

Sample Output

3, 2, 7, 6, 8

[Hint:

First, count how many digits. Then calculate 10 to the power that (number of digits) minus 1.

Say, 32768 has 5 digits, so you calculate 10 to the power 4 which is 10,000.

Then divide 32,768 by 10,000 and thus you get 3

Take remainder of 32,768 by 10,000 and thus you get 2,768

Then divide 10,000 by 10 to get 1,000

Then divide 2,768 by 1,000 and thus you get 2.

take remainder of 2,768 by 1,000 and thus you get 768

keep going on until there are no more digits left (zero!).

In short:

Part 1: First count digits, say 5 in this case for 32,768

Part 2: Then calculate 10 to the power 4 (5-1), that is 10,000.

Part 3: Then repeat the following three steps:

$$32,768 / 10,000 = 3$$

$$32,768 \% 10,000 = 2,768$$

$$10,000/10 = 1,000$$

$$2,768 / 1,000 = 2$$

$$2,768 \% 1,000 = 768$$

$$1,000/10 = 100$$

$$768 / 100 = 7$$

$$768 \% 100 = 68$$

$$100/10 = 10$$

$$68 / 10 = 6$$

$$68 \% 10 = 8$$

$$10/10 = 1$$

$$8 / 1 = 8$$

$$8 \% 1 = 0$$

$$1/10 = 0]$$

6. Write a Java program that will take an integer as input and -

a) Find out if the number is a prime number or not.

b) Find out if the number is a perfect number or not.

[Prime Number: If a number has **only two divisors**, (1 and itself), then it is a prime number. Else, then it is not a prime number.

Perfect Number: A number is said to be a perfect number if the **sum of its divisors**, including 1 but not the number itself is equal to that number.]

Sample Input

6

Sample Output

6 is not a prime number

6 is a perfect number