

No. of Pages	4
No. of Questions	4
Total Marks	25
Time: 70 Minutes	

## Department of Computer Science and Engineering

### Midterm Examination

### CSE 110: Programming Language I

---

- ❖ Write theory teacher's Initial on top of the answer script in LARGE FONT.
- ❖ Answer all questions. Use **back part** of the answer script for rough work.
- ❖ Answer Question **1 & 2** at the **beginning part** of the answer script.
- ❖ Figure in bracket [] next to each question indicates marks for that question.
- ❖ At the end of exam, put **question paper** inside answer script and **return both**.
- ❖ Understanding the question is part of the exam, **please do not ask questions**.

No washroom breaks

Section: \_\_\_\_\_ ID: \_\_\_\_\_ Name in CAPITAL: \_\_\_\_\_

#### **Question 1 [CO1] [10 Points]**

**[Answer on the answer script]**

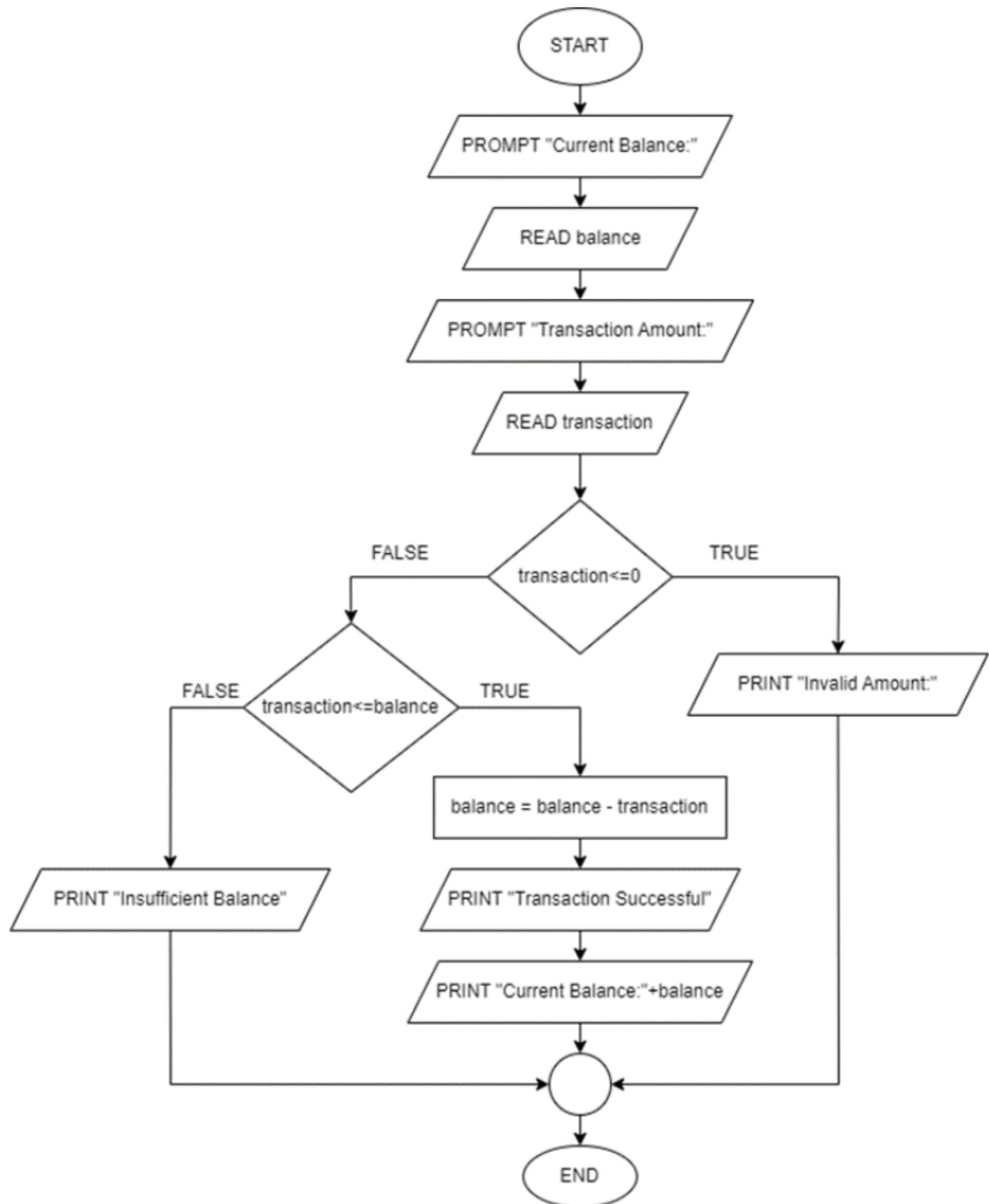
While wandering through the Amazon, you found a hidden temple. Upon entering the temple, you find a treasure chest which requires you to solve a riddle to open. The chest asks for two numbers which will define the range. Between the range, there must be composite numbers. The count of composite numbers should be a factor of the absolute difference between the start and end value of the range. If you can provide such range, the treasure is yours. Since you are confident in your coding skills, you decide to write a Java program on your laptop which will help you test out some ranges before entering the numbers in the treasure chest.

Now write a Java program that takes two integers (start and end) as inputs from the user and finds out how many composite numbers are inside the range. If the absolute difference between start and end value is a multiple of the composite number count, print "Congratulations! The treasure is yours". If the count is not a multiple of the mentioned absolute difference, print "Close! But not quite". If there are no composite numbers in the range, print "Too far off."

Sample Input	Sample Output	Explanation
Start: 8 End: 12	8 9 10 12 Congratulations! The treasure is yours.	In the range 8 to 12, there are 4 numbers (8,9,10,12) which have more than 2 divisors. The absolute difference between the ranges is (12 - 8 ) = 4. Since 4 is divisible by 4, the treasure was found.
Start: 5 End: 14	6 8 9 10 12 14 Close! But not quite.	In the range 5 to 14, there are 6 numbers (6,8,9,10,12,14) which have more than 2 divisors. The absolute difference between the ranges is (14 - 5 ) = 9. Since 9 is not divisible by 6, the treasure was not found.
Start: 1 End: 3	Too far off.	There are no composite numbers in this range

**Question 2 [CO1] [6 Points]**  
[Answer on the answer script]

Implement a Java program from the given flowchart-



**Question 3 [CO4] [4 Points]**  
**[Answer on the question paper]**

The code below is designed to print an integer number from right to left. However, it contains 5 errors. Analyze the code, identify the errors, and correct them. Provide your corrections in the specified format.

1	import java.util.Scanner;
2	public class test
3	{
4	public static void main (String args)
5	{
6	Scanner scan = new Scanner (System);
7	int num = scan.nextInt();
8	int temp1 = '0';
9	while(num == 0)
10	{
11	temp1 = num / 10;
12	num /= 10;
13	System.out.print(temp1 + " ");
14	}
15	}
16	}

Write your corrections in the table below. For better understanding, 1 error correction is shown.

Line Number	Fix
Line 4	public static void main (String [] args)

**Question 4 [CO1] [5 Points]**  
[Answer on the question paper]

Illustrate the outputs of the following statements. Provide your workings on the answer script to verify your outputs. Your answer will not be accepted without the workings. All of the outputs must be in the question paper.

1	public class TracingA{	Output
2	public static void main(String[] args) {	
3	int n1 = 184; int n2 = 24; int n3 = 56;	
4	while(! ((n1 == n2) && (n2 == n3))){	
5	if (n1>n2){	
6	if (n2>n3){	
7	n1 -= n2;	
8	System.out.println(n1);	
9	}	
10	else{	
11	if (n1>n3){	
12	n1 -= n3;	
13	System.out.println(n1);	
14	}	
15	else{	
16	n3 -= n1;	
17	System.out.println(n3);	
18	}	
19	}	
20	}	
21	else{	
22	if (n2>n3){	
23	if (n1>n3){	
24	n2 -= n1;	
25	System.out.println(n2);	
26	}	
27	else{	
28	n2 -= n3;	
29	System.out.println(n2);	
30	}	
31	}	
32	else{	
33	n3 -= n2;	
34	System.out.println(n3);	
35	}	
36	}	
37	}	
38	System.out.println(n1);	
39	}	
40	}	