BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Midterm
Duration: 75 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Fall 2024
Full Marks: 25
No. of Pages: 2

Name: Tanzeebul Tamim (Please write in CAPITAL LETTERS)	ID: 23301265	Section: 02

A

✓ Use the back **part** of the answer script for rough work. **No washroom breaks.**
✓ At the end of the exam, put the question **paper** inside the answer script and **return both.**

Question 1: CO2 [7 Points]

```
1  public class MidA{
2    public int temp = 3, z = 2, sum;
3    public void m1(){
4      int x = 3, temp = -1;
5      x = sum + 4 + z++;
6      temp += this.temp;
7      sum = this.temp + m2(sum, x) + this.temp;
8      System.out.println(x +" "+ temp+" "+ sum);
9    }
10   public int m2(int x, int sum){
11     temp = x;
12     int temp = 6;
13     temp = ++this.temp + z + this.z;
14     x = sum + (temp++) + temp;
15     sum = this.sum + x + this.temp;
16     System.out.println(x +" "+ temp+" "+ sum);
17     return this.temp--;
18   }
19 }
```

Illustrate the output of the following statements written in main method of a tester class. [Answers without workings on the script will be rejected]:

```
MidA t1 = new MidA();
t1.m1();
t1.m2(3, 5);
```

Output [Answer on question paper]		
21	8	22
6	2	4
26	11	34

Question 2: [8 Points]

A teacher wants to evaluate the performance of students based on their marks. Instead of calculating the average marks for the entire section, the teacher is interested in finding the maximum average marks among some groups of k consecutive students. Given an array of student's marks and a value k, write a Java program that calculates and prints the maximum average among groups of k consecutive student's marks. Your code should work for any given array of positive integers.

[**NOTE:** You can assume the value of k will always be less than the length of the array]

Given Array & Sample Input	Output	Explanation

`int [] marks = {12,15,20,17,20,13};`
`Enter value of k: 4`
	`Max Avg: 18.0`	`(12+15+20+17)/4 = 16.0`
		`(15+20+17+20)/4 = 18.0`
		`(20+17+20+13)/4 = 17.5`
		`Among 16.0,18.0 and 17.5, 18.0 is the maximum average.`

int [] marks = {13,11,19,17,16,15,18,11}; Enter value of k: 7	Max Avg: 15.57	(13+11+19+17+16+15+18)/7 = 15.57 (11+19+17+16+15+18+11)/7 = 15.28 Among 15.57 and 15.28, **15.57** is the maximum average.

Question 3: CO1 [10 Points]

Design a class called **Boxer** with appropriate attributes and features so that the driver code below generates the expected output.

[NOTE: `winLossRatio()` should calculate the ratio of fights won to fights lost, based on the total number of fights attempted.]

Driver Code	Expected Output

public class BoxerTest {
public static void main(String[] args) {
Boxer jake = new Boxer();
jake.updateDetails("Jake", 102.95, 3);
System.out.println("========1========");
jake.showDetails();
System.out.println("========2========");
jake.fight("Deji", true);
System.out.println("========3========");
jake.showDetails();
jake.fight("Tommy", false);
System.out.println("========4========");
System.out.println(jake.winLossRatio());
System.out.println("========5========");
jake.fight("Felix");
jake.showDetails();
System.out.println("========6========");
jake.fight("Mike", true);
System.out.println("========7========");
jake.showDetails();
System.out.println("========8========");
System.out.println(jake.winLossRatio());
}
}
	Jake can fight 3 opponent(s)
	========1========
	Name: Jake, Weight: 102.95 KG
	No fights yet.
	========2========
	========3========
	Name: Jake, Weight: 102.95 KG
	Fight history:
	vs Deji: Won
	========4========
	Career stats: 1:1
	========5========
	Name: Jake, Weight: 102.95 KG
	Fight history:
	vs Deji: Won
	vs Tommy: Lost
	vs Felix: Lost
	========6========
	Cannot fight more than 3 opponent(s)
	========7========
	Name: Jake, Weight: 102.95 KG
	Fight history:
	vs Deji: Won
	vs Tommy: Lost
	vs Felix: Lost
	========8========
	Career stats: 1:2