DO NOT WRITE ANYTHING ON QPAPER EXCEPT ROLL NO.

CHITKARA UNIVERSITY

2022-2023

Sessional Test I – April, 2023

Roll No: ……………… [Total No. of Pages: 8]

Programme: B.E. (CSE) **Time: 90 minutes**

Course Title: Core Java

Course Code: CS109 **Max. Marks: 40**

General Instructions:

- Follow the instructions given in each section.

Section – A

(Q 1 to 10: Each Question carries 1 mark)

Q1. Which of these is a correct statement about args in the following line of code?
public static void main(String args[])

a. args is a String
b. args is a Character
c. args is an array of String
d. args is an array of Character

Q2. Which of these data types is used to store command line arguments?

a. Array
b. Stack
c. String
d. Integer

Q3. Which keyword is used to refer to the current object in Java?

a. this
b. self
c. current
d. Object

Q4. When is the object created with new keyword?

a. At compile time
b. Depends upon the code
c. At run time
d. None of the above

Q5. Automatic type conversion is possible in which of the possible cases?

a. long to float
b. int to long
c. long to int
d. Short to byte

Q6. Which of these is necessary to specify at time of array initialization?

a. Row
b. Column
c. Both Row and Column

d. None of the mentioned

Q7. The order of calling constructor in case of inheritance is

a. Super class constructor and then Subclass Constructor
b. Subclass constructor and then Super class Constructor
c. Super class and subclass constructors are independent
d. Super class and subclass constructors can be called in any order

Q8. Which of these statements are incorrect?

a. Equal to operator has least precedence
b. Brackets () have highest precedence
c. Division operator, /, has higher precedence than multiplication operator
d. Addition operator, +, and subtraction operator have equal precedence

Q9. What is the value of "age" in the below Java program with a DO-WHILE loop?

```
int age=20;
do
{
  age++;
}while(age<20);
System.out.println(age);
```

a. 20
b. 21
c. Compiler error
d. None

Q10. What will be the output of the following program?

```
class jump_statments
 {
    public static void main(String args[])
    {
        int x = 2;
        int y = 0;
        for ( ; y < 10; ++y)
        {
            if (y % x == 0)
                continue;
            else if (y == 8)
                 break;
            else
               System.out.print(y + " ");
        }
    }
 }
```

a. 1 3 5 7
b. 2 4 6 8
c. 1 3 5 7 9
d. 1 2 3 4 5 6 7 8 9

Section - B
(Q 11 to 15 : Each Question carries 2 marks)

Q11. What will be the output of following code?

```
public class main{
public static void main(String [] args){
          int [] a= new int [0];
          System.out.println(a.length);
                         }
                         }
```

a. 0
b. Compilation Error
c. Runtime Error
d. None of the these

Q12. What will be the output of the following program?

```
public class Test {
public static void main(String[] args) {
int count = 1;
while (count<=15) {
System.out.println(count%2==1?"***":"+++++");
++count;
} // end while
} // end main
}
```

a. 8 times * and 7 times +++++
b. 15 times *
c. 15 times +++++
d. Both will print only once

Q13. Which of the following is not an advantage to using inheritance?

a. Similar classes can be made to behave consistently.
b. One big superclass can be used instead of many little classes.
c. Code that is shared between classes needs to be written only once.
d. Enhancements to a base class will automatically be applied to derived classes.

Q14. Which code line could possibly "call" this method?

```
public static int SomeMethod(double[ ] array, int[ ] number)
{
. . .
}
```

a. int value = SomeMethod(money, grades);
b. SomeMethod(money, grades);
c. double value = SomeMethod(money, grades);
d. int value = SomeMethod(money);

Q15. What is the output of the following code?

```
int x = 10;

while (x > 0) {
```

```
System.out.print(x);

x--;

}
```

- **a. 10 9 8 7 6 5 4 3 2 1**
- b. 1 2 3 4 5 6 7 8 9 10
- c. 0
- d. Compilation error

Section - C

(Q 16 to 17: Each Question carries 5 marks)

Q16. Take an input N, the size of array. Take N more inputs and store that in an array. Write a function which returns the maximum value in the array. Print the value returned.

1.It reads a number N.

2.Take Another N numbers as input and store them in an Array.

3.calculate the max value in the array and return that value.

Input Format

First line contains integer n as size of array. Next n lines contain a single integer as element of array.

Constraints

N cannot be Negative. Range of Numbers can be between -1000000000 to 1000000000

Output Format

Print the required output.

Sample Input

```
4
2
8
6
4
```

Sample Output

```
8
```

Explanation

Arrays= {2, 8, 6, 4} => Max value = 8.

SOLUTION:

```
import java.util.*;
public class Main {
public static void main(String args[]) {
Scanner sc=new Scanner(System.in);
int n=sc.nextInt();
int arr[]=new int[n];
for(int i=0;i<n;i++){
arr[i]=sc.nextInt();
}
int max=Integer.MIN_VALUE;
for(int i:arr){
max=Math.max(max,i);
}
System.out.println(max);
}
}
```

Test Cases:

Input	**Output**
6 -101 -102 -105 -107 -99 -200002	-99
1 18	18
9 6 0 -2 4 -8 1 7 16 8	16
2 45 -45	45
2 45 80	80

Q17: Working with 2D arrays is quite important. Here we will do swapping of columns in a 2D array. You are given a matrix M or r rows and c columns. You need to swap the first column with the last column.

Input format:

First line contains two integers n and m.
Next lines contain n*m matrix.

Output format:

Print the matrix after modification.

Sample Input:

```
3 4
1 2 3 4
4 3 2 1
6 7 8 9
```

Sample Output:

```
4 2 3 1
1 3 2 4
9 7 8 6
```

	Test Case 1	**Test Case 2**	**Test Case 3**
Input	2 2 1 2 3 4	3 4 1 2 3 4 5 6 7 9 1 2 1 2	4 4 1 2 3 4 4 3 2 1 6 5 7 8 1 2 3 4
Output	2 1 4 3	4 2 3 1 9 6 7 5 2 2 1 1	4 2 3 1 1 3 2 4 8 5 7 6 4 2 3 1

Solution:

```
import java.util.Scanner;
class Main{
        static void solve(int a[][],int r, int c){
        for(int i = 0;i<r;i++){
        int temp = a[i][0];
        a[i][0] = a[i][c-1];
        a[i][c-1] = temp;

        }
        for(int i = 0;i<r;i++){
        for(int j = 0;j<c;j++){
          System.out.print(a[i][j] + " ");
        }
        System.out.println();
        }
        }
        public static void main(String arg[])
        {
        int n,m;
        Scanner sc = new Scanner(System.in);
        n=sc.nextInt();
        m=sc.nextInt();
        int a[][] = new int[n][m];
        for(int i=0;i<n;i++)
        {
        for(int j=0;j<m;j++)
        a[i][j] = sc.nextInt();
        }
        solve(a,n,m); }   }
```

Section - D

(Q 18: Question carries 10 marks)

Q18: Write a Java program to print a Water Image Mirror form of Star Dot pattern for a size of N.

Input format:

The first line of the input contains the size of the pattern.

Constraints:

1 <= N <= 100

Output format:

The output will contain the mirror form of N sized pattern using star(*) and dot(.)

Sample Input:

```
 4
```

Sample Output:

```
...*
..**
.***
****
****
.***
..**
...*
```

Explanation: -Use the dot(.) character in place of whitespace before the first star(*) character of any row.

There is no space between star(*) characters.

There is no space after the last star(*) character.

SOLUTION:

```
import java.util.*;
public class Main {
        private static void displayUpperPart(int size)
        {
        int m, n;
        for (m = size - 1; m >= 0; m--) {
        for (n = 0; n < m; n++) {
           System.out.print(".");
        }
        for (n = m; n <= size - 1; n++) {
           System.out.print("*");
        }
        System.out.println();
        }
        }
        private static void displayLowerPart(int size)
        {
        int m, n;
        for (m = 1; m <= size; m++) {
        for (n = 1; n < m; n++) {
           System.out.print(".");
        }
        for (n = m; n <= size; n++) {
           System.out.print("*");
        }
                System.out.println();
        }
        }
        public static void main(String[] args)
        {
        Scanner sc = new Scanner(System.in);
        int size = sc.nextInt();
        displayUpperPart(size);
        displayLowerPart(size);
        sc.close();
        }
}
```

Test case 1	**Test case 2**	**Test case 3**	**Test Case 4**	**Test case 4**
Input 2	Input 3	Input 5	Input 6	Input 7
Output .* ** ** .*	**Output** ..* .** *** *** .** ..*	**Output** ....* ...** ..*** .**** ***** ***** .**** ..*** ...** ....*	**Output** .....* ....** ...*** ..**** .***** ****** ****** .***** ..**** ...*** ....** .....*	**Output** ......* .....** ....*** ...**** ..***** .****** ******* ******* .****** ..***** ...**** ....*** .....** ......*