

**Sessional Test I – April, 2023**

Roll No: .....

[Total No. of Pages: 9]

**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 the following for loop declaration is not valid?

- a. **for ( int i = 99; i >= 0; i / 9 )**
- b. **for ( int i = 20; i >= 2; i++ )**
- c. **for ( int i = 2; i <= 20; i = 2\*i )**
- d. **for ( int i = 7; i <= 77; i += 7 )**

**Q2.** What are the types of memory allocated in memory in java?

- a. Heap memory
- b. Stack memory
- c. **Both Heap and Stack memory**
- d. None of these

**Q3.** What is the output of the program?

```
public class output{  
    public static void main(String[] args){  
        if(1+1+1+1+1==5){  
            System.out.println("True");  
        }  
        else{  
            System.out.println("False");  
        }  
    }  
}
```

- a. false
- b. Compile Error
- c. **true**
- d. None

**Q4.** Which one is a valid declaration of a Boolean?

- a. **boolean b2 = 'false';**
- b. **boolean b3 = false;**
- c. **boolean b4 = Boolean.false();**
- d. **boolean b5 = no;**

**Q5.** What is the output of the program?

```
public class CppBuzz {  
    public static void main(String[] args){  
        int a = 5+5*2+2*2+(2*3);  
        System.out.println(a);  
    }  
}
```

- a. 110
- b. 10
- c. 25**
- d. 13

**Q6.** Find the output of the following code.

```
public class Solution{  
    public static void main(String[] args){  
        int x = 5;  
        x*=(2+7);  
        System.out.println(x);  
    }  
}
```

- a. 22
- b. 45**
- c. 10
- d. None

**Q7.** What is the output of the program?

```
public class Solution{  
    public static void main(String[] args){  
        int[] x={120,200,016};  
        for(int i=0;i<x.length;i++){  
            System.out.print(x[i] + " ");  
        }  
    }  
}
```

- a. 120 200 016
- b. 120 200 14**
- c. 120 200 16
- d. None

**Q8.** Java Array can allocate \_\_\_\_.

- a. Dynamic Memory
- b. Static Memory**
- c. Both A and B
- d. None of the above

**Q9.** Which of the following is a superclass of every class in Java?

- a. ArrayList
- b. Abstract class
- c. Object class**
- d. String

**Q10.** What is the output of the following code?

```
int x = 5;  
System.out.println(++x);
```

- a. 5
- b. 6**
- c. Compilation error
- d. Runtime error

**Section - B**

*(Q 11 to 15: Each question carries 2 marks)*

**Q11.** What is the output?

```
public class Main {  
    public static void main(String arg[])  
    {
```

```
        int i;  
        for (i = 1; i <= 12; i += 2)  
        {  
            if (i == 8)  
            {  
                System.out.println(i);  
                break;  
            }  
        }  
    }  
}
```

- a. 1
- b. No output**
- c. 8
- d. 1357911

**Q12.** What will be the output of the following code?

```
public class Main {  
    public static void main(String args[]) {  
        int [] x= { 1,2,3,4};  
        int [] y=x;  
        x=new int[2];  
        for(int i=0;i<x.length;i++) {  
            System.out.print(y[i]+" ");  
        }  
    }  
}
```

- a. 1 2 3 4
- b. 0 0 0 0
- c. 1 2**
- d. 0 0

**Q13.** What will be the output of the following code?

```
public class Main {  
    public static void main(String args[]) {  
        int [] x= { 12,14,4,7};
```

```
int [] y=x;
x=new int[4];
for(int i=0;i<x.length;i++) {
    System.out.print(y[i]+" ");
}
}
```

- a. 12 14 4
- b. 0 0 0 0
- c. **12 14 4 7**
- d. 0 0 0

**Q14.** What is the output of this program?

```
class access{
    public int x;
    private int y;
    void cal(int a, int b){
        x = a + 1;
        y = b;
    }
}
class access_specifier {
    public static void main(String args[])
    {
        access obj = new access();
        obj.cal(2, 3);
        System.out.println(obj.x + " " + obj.y);
    }
}
```

- a. 3 3
- b. 2 3
- c. Runtime Error
- d. **Compilation Error**

**Q15.** What will happen when you attempt to compile and run the following code?

```
int Output = 10;
boolean b1 = false;
if((b1 == true) && ((Output += 10) == 20))
{
    System.out.println("We are equal " + Output);
}
else
{
    System.out.println("Not equal! " + Output);
}
```

- a. Compilation error, attempting to perform binary comparison on logical data type.
- b. Compilation and output of "We are equal 10".
- c. Compilation and output of "Not equal! 20".
- d. **Compilation and output of "Not equal! 10".**

**Section - C**  
(Q 16 to 17: Each question carries 5 marks)

**Q16.** Given an array nums of length n. We define a running sum of an array as  $\text{runningSum}[i] = \text{sum}(\text{nums}[0] \dots \text{nums}[i])$ .

Return the running sum of nums.

**Input Format**

First line contains an integer n representing number of elements. Next line contains n integers denoting array elements.

**Constraints**  $1 \leq \text{nums.length} \leq 1000$   
 $-10^6 \leq \text{nums}[i] \leq 10^6$

**Output Format**

An integer representing running sum array of the given array

**Sample Input**

4

1 2 3 4

**Sample Output**

1 3 6 10

**Explanation**

Running sum is obtained as follows: [1, 1+2, 1+2+3, 1+2+3+4].

Test Cases:

Input	Output
4 1 2 3 4	1 3 6 10
3 1 2 3	1 3 6
1 1	1
5 1 3 4 5 6	1 4 8 13 21
2 1 3	1 4

**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 pref[]=new int[n];
        pref[0]=arr[0];
        System.out.print(pref[0]+" ");
```

```

for(int i=1;i<n;i++){
pref[i]=pref[i-1]+arr[i];
System.out.print(pref[i]+" ");
}}

```

**Q17:** Suppose you are working on a project that requires you to create a program that takes an integer input from the user and then reverses it. Your program should be able to handle any integer value provided by the user. Your task is to write a program that accepts an integer N and reverses it. The reversed integer should be printed to the console.

**Input format:**

The first line contains an integer T, total number of test cases. Then follow T lines, each line contains an integer N.

**Output format:**

For each test case, display the reverse of the given number N, in a new line.

**Sample Input:**

```

2
12345
31203

```

**Sample Output:**

```

54321
30213

```

	Test Case 1	Test Case 2	Test Case 3
Input	3 2123 2300 678	2 1101 23	4 1234 4321 1 10
Output	3212 32 876	1011 32	4321 1234 1 1

**Solution:**

```

public class Main{
    public static void main(String[] dtr)
    {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while(t-- >0)
        {
            // Scanner sc = new Scanner(System.in);
            int a = sc.nextInt();
            int ans=0;
            while(a>0)
            {
                ans = ans*10 + a%10;
                a=a/10;
            }
        }
    }
}

```

```

    }
    System.out.println(ans);
    }
}

```

**Section - D**  
(Q 18: Question carries 10 marks)

**Q18:** You are provided with three numbers: input1, input2, and input3. Each of these is a four-digit number within the range  $\geq 1000$  and  $\leq 9999$ , i.e.  $1000 \leq \text{input1} \leq 9999$ ,  $1000 \leq \text{input2} \leq 9999$ ,  $1000 \leq \text{input3} \leq 9999$ . Write a JAVA program to find a Key using the below formula:

Key = [SMALLEST digit in the thousands place of all three numbers] [LARGEST digit in the hundreds place of all three numbers] [SMALLEST digit in the tens place of all three numbers] [LARGEST digit in the units place of all three numbers].

**Input Format**

The input consists of three space-separated four-digit integers - input1, input2, and input3.

**Output Format**

Print an integer representing the concatenation of four digits where the first digit from the left represents the smallest digit in the thousands place of all three numbers; the second digit represents the largest digit in the hundreds place of all three numbers; the third digit represents the smallest digit in the tens place of all three numbers, and the fourth digit represents the largest digit in the unit place of all three numbers.

**Sample Input 1**

3521 2452 1352

**Sample Output 1**

1522

**Explanation**

Key = [smallest digit in the thousands place of all three numbers] [LARGEST digit in the hundreds place of all three numbers] [smallest digit in the tens place of all three numbers] [LARGEST digit in the units place of all three numbers]

If input1 = 3521, input2=2452, input3=1352, then Key = [1][5][2][2] = 1522.

**Sample Input 2**

1234 4321 1243

**Sample Output 2**

1324

	Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5
Input	3212 3425 2349	5783 2341 2153	3421 2423 1234	1111 2222 3333	3521 2452 1352
Output	2419	2743	1424	1313	1522

**Solution**

```
import java.util.*;
class Main
{
    public static int fun(int a, int b, int c)
    {
        int n = 1, temp = 0;
        for(int i = 0; i < 4; i++)
        {
            int rem = 0, large = -999, small = 999;
            if(i % 2 == 0)
            {
                rem = a % 10;
                a = a / 10;
                if(rem > large)
                {
                    large = rem;
                }
                rem = b % 10;
                b = b / 10;
                if(rem > large)
                {
                    large = rem;
                }
                rem = c % 10;
                c = c / 10;
                if(rem > large)
                {
                    large = rem;
                }
                temp = (large * n) + temp;
            }
            else
            {
                rem = a % 10;
                a = a / 10;
                if(rem < small)
                {
                    small = rem;
                }
                rem = b % 10;
                b = b / 10;
                if(rem < small)
                {
                    small = rem;
                }
                rem = c % 10;
                c = c / 10;
                if(rem < small)
                {
                    small = rem;
                }
                temp = (small * n) + temp;
            }
            n = n * 10;
        }
    }
}
```



```
return temp;
}
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
int b = sc.nextInt();
int c = sc.nextInt();
System.out.print(fun(a, b, c));
}
}
```