OOP_JAVA_CIA_1_ReTest

Test Summary

- No. of Sections: 2
- No. of Questions: 11
- Total Duration: 90 min

Section 1 - MCQ

Section Summary

- No. of Questions: 9
- Duration: 15 min

Additional Instructions:

None

Q1. Consider the following code snippet and answer the question that follows.

Which of the following statements is correct?

```
public class Vehicle
{
   . . .
   public void setVehicleAtrributes()
   {
      . . .
   }
}
public class Auto extends Vehicle
{
   . . .
   public void setVehicleAtrributes()
   {
      . . .
   }
}

```

The subclass is shadowing a superclass method.

The subclass is overloading a superclass method.

The subclass is overriding a superclass method.

This code will not compile.

Q2. What will be the output of the following code snippet?

```
ArrayList al = new ArrayList();
al.add(30);
al.add(20);
al.add(10);
al.add(40);
al.add(50);
Collections.sort(al, Collections.reverseOrder());
System.out.println(al);

```

[10,20,30,40,50]

[50,40,10,20,30]

[50,40,30,20,10]

None

Q3. How can we remove an object from ArrayList?

remove() method

using Iterator

remove() method and using Iterator

delete() method

Q4. What will be stored in the object obj in the following line of code ?
ItemType obj;

Memory address of allocated memory of object

NULL

Any arbitrary pointer

Garbage

Q5. Which among the following best defines multilevel inheritance?

A class derived from another derived class

Classes being derived from other derived classes

Continuing single level inheritance

Class which have more than one parent

Q6. Which of the following is true about class Object.
I. The class Object is a superclass of all other classes.
II. A variable of type Object can hold reference to any object or a null reference.
III. You must explicitly extend class Object.
IV. All class and array types inherit the methods of a class Object.

I and II

I, II and III

I, II and IV

I and IV

Q7. Which of the following is not a reason to use encapsulation when designing a class?

Promote usability by other developers.

Maintain class data integrity of data elements.

Prevent users from modifying the internal attributes of a class.

Increase concurrency and improve performance.

Q8. What is true about protected constructor?

Protected constructor can be called directly

Protected constructor can only be called using super()

Protected constructor can be used outside package

protected constructor can be instantiated even if child is in a different package

Q9. Consider the hierarchy of classes shown below.

Which represent valid class headers that would be found in this hierarchy?

public class ScriptedShow extends TelevisionShow {. . .
public class Comedy extends ScriptedShow {. . .

public class TelevisionShow extends ScriptedShow {. . .
public class ScriptedShow extends Comedy {. . .

public class Drama extends TelevisionShow {. . .
public class Comedy extends Drama {. . .

public class ScriptedShow extends RealityShow {. . .
public class RealityShow extends ScriptedShow {. . .

Section 2 - Coding

Section Summary

- No. of Questions: 2
- Duration: 75 min

Additional Instructions:

None

Q1. Input a positive integer N (N > 0), input N strings, and sort the strings in place in the order of increasing length. Print the sorted strings using ArrayList as an implementation of the List interface for storing the individual strings.

Input Format

Input number of elements

Input each string on a separate line

Output Format

Print the list of strings sorted by their length

Sample Input

```
3
aa
b
ccc
```

Sample Output

```
[b, aa, ccc]
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. **Problem Statement**

Complete the snippet to check whether the given two numbers(user inputs) are Friendly or not.

Explanation

Two numbers are said to be friendly if their sum of digits(SOD) reduced up to a single digit is equal otherwise they are not friendly.

Example for the SOD of a number reduced up to single digit

Assume the number 12345
Its SOD reduced up to a single digit will be as follows.
12345 = 1+2+3+4+5 = 15 = 1+5 = 6

Note: For a better understanding, refer to the **HINT** section.

Input Format

Two integer inputs

Output Format

Refer to the sample output.

Constraints

Numbers are only positive.

Sample Input

```
12345
52413
```

Sample Output

```
SOD of a reduced upto single digit is : 6
SOD of b reduced upto single digit is : 6
So they are Friendly
```

Sample Input

```
12346
54321
```

Sample Output

```
SOD of a reduced upto single digit is : 7
SOD of b reduced upto single digit is : 6
So they are Not friendly
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - MCQ

Q1

The subclass is overriding a superclass method.

Solution

No Solution

Q2

[50,40,30,20,10]

Solution

No Solution

Q3

remove() method and using Iterator

Solution

There are 2 ways to remove an object from ArrayList. We can use overloaded method remove(int index) or remove(Object obj). We can also use an Iterator to remove the object.

Q4

NULL

Solution

No Solution

Q5

Classes being derived from other derived classes

Solution

No Solution

Q6

I, II and IV

Solution

No Solution

Q7

Increase concurrency and improve performance.

Solution

No Solution

Q8

Protected constructor can only be called using super()

Solution

No Solution

Q9

public class ScriptedShow extends TelevisionShow {. . .

public class Comedy extends ScriptedShow {. . .

Solution

No Solution

Section 2 - Coding

Q1

Test Case

Input

```
3
111
22
3333
```

Output

```
[22, 111, 3333]
```

Weightage - 80

Input

```
1
3a
```

Output

```
[3a]
```

Weightage - 10

Input

```
3
23
111
0000
```

Output

```
[23, 111, 0000]
```

Weightage - 10

Sample Input

```
3
aa
b
ccc
```

Sample Output

```
[b, aa, ccc]
```

Solution

```
import java.util.*;
import java.lang.*;
import java.io.*;

class Q01Simple_Sort
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner input = new Scanner(System. in);
        // Input number of elements
        int number_of_elements = input. nextInt();
        input. nextLine();
        if (number_of_elements <= 0) return;
        List<String> list = new ArrayList<>();
        for (int ctr = 0; ctr < number_of_elements; ctr++) {
            // Input next string
            String str = input. nextLine();
            list.add(str);
        }
        Collections.sort(list, new Comparator<String>() {
            public int compare(String o1, String o2) {
            return o1.length() - o2.length();
        }});
        System.out.println(list);
    }
}
```

Q2

Test Case

Input

```
12345677
98765432
```

Output

```
SOD of a reduced upto single digit is : 8
SOD of b reduced upto single digit is : 8
So they are Friendly
```

Weightage - 25

Input

```
12345677
98765431
```

Output

```
SOD of a reduced upto single digit is : 8
SOD of b reduced upto single digit is : 7
So they are Not friendly
```

Weightage - 25

Input

```
12345676
98765431
```

Output

```
SOD of a reduced upto single digit is : 7
SOD of b reduced upto single digit is : 7
So they are Friendly
```

Weightage - 25

Input

```
12345676
98765430
```

Output

```
SOD of a reduced upto single digit is : 7
SOD of b reduced upto single digit is : 6
So they are Not friendly
```

Weightage - 25

Sample Input

```
12345
52413
```

Sample Output

```
SOD of a reduced upto single digit is : 6
SOD of b reduced upto single digit is : 6
So they are Friendly
```

Sample Input

Sample Output

```
12346
54321
```

```
SOD of a reduced upto single digit is : 7
SOD of b reduced upto single digit is : 6
So they are Not friendly
```

Solution

Header

```
// You are using Java
import java.util.Scanner ;
class Main{

    static class A{

        private int a ;
        private int b ;




        A(int c, int d){
            a = c ;
            b = d ;
        }

        boolean checkFriendly(){
            System.out.println("SOD of a reduced upto single digit is : " + a%9) ;
            System.out.println("SOD of b reduced upto single digit is : " + b%9) ;
            return (a%9==b%9) ;
        }
    }
```

Footer

```
    public static void main(String []args){

        Scanner sc = new Scanner(System.in) ;

        int a = sc.nextInt() ;
        int b = sc.nextInt() ;
        A o = new A(a, b) ;
        if(o.checkFriendly()){
            System.out.print("So they are Friendly") ;
        }
        else{
            System.out.print("So they are Not friendly") ;
        }
    }
}
```