OOPS_USING_JAVA_RETEST_CIA1

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. In the below java code, whose “Car” will be called?

```
class Father {

   public void car() {
      System.out.println("Father's Car");
   }
}

class Son extends Father {

   public void car() {
      System.out.println("Son's Car");
   }
}

public class Sample {

   public static void main(String[] args) {

      Son john = new Son();
      john.car();
   }

}

```

Father’s Car

Son’s Car

There is an ambiguity, so no one's car will be called

Compiler Error

Q2. What will be the output of the following Java program?

```
class A
{
   int i;
}
class B extends A
{
   int j;
   void display()
   {
      super.i = j + 1;
      System.out.println(j + " " + i);
   }
}
class inheritance
```

```
{
   public static void main(String args[])
   {
      B obj = new B();
      obj.i=1;
      obj.j=2;
      obj.display();
   }
}
```

2 2

3 3

2 3

3 2

Q3. How will you call a static method of an interface in a class?

Using super keyword along with interface name.

Using name of the interface.

Both of the above.

None of the above.

Q4. What is the expected output?

```
public class profile
{
   private profile(int w)//line 1
   {
      System.out.println(w);
   }
   public static profile() //line 5
   {
      System.out.println(10);
   }
   public static void main(String[] args)
   {
      profile obj=new profile(50);
   }
}
```

10 50

Won't compile because of line (1), constructor can't be private

50

Won't compile because of line (5), constructor can't be static

Q5. In a Multi-Level Inheritance in Java, the last subclass inherits methods and properties of ________.

Only one immediate Superclass

Few classes above it.

All classes above it

None of the above

Q6. Choose the correct option based on this program:

```
import java.util.*;
class UtilitiesTest {
    public static void main(String[] args) {
        List < int > intList = new ArrayList < > ();
        intList.add(10);
        intList.add(20);
        System.out.println("The list is: " + intList);
}
}
```

It prints the following: The list is: [10, 20]

It prints the following: The list is: [20, 10]

It results in a compiler error

It results in a runtime exception

Q7. Given below code snippet
Which of the following are valid statements?

```
interface A{
int aMethod(String s);
}
```

A a =a -> a.length();

A x = y -> {return y;};

A s = "2" -> Integer.parseInt(s);

A b = (String s) -> 1;

Q8. What concepts come under Polymorphism in java?
1.Method overloading
2.Constructor overloading
3.Method overriding
4.All the above

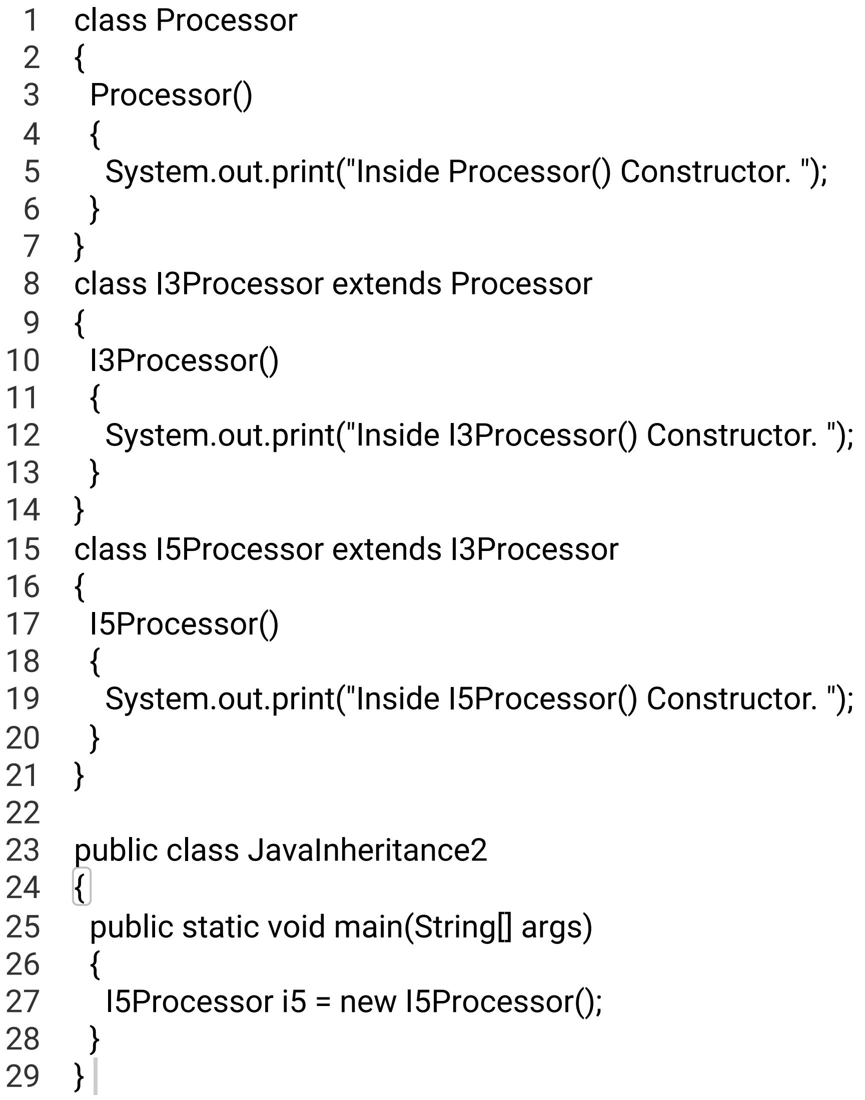
option 1 and 2

option 2 and 3

only 3

option 4

Q9. What is the output of the below java program with Constructors and Inheritance?



```
class Processor
{
 Processor()
 {
  System.out.print("Inside Processor() Constructor. ");
 }
}
class I3Processor extends Processor
{
 I3Processor()
 {
  System.out.print("Inside I3Processor() Constructor. ");
 }
}
class I5Processor extends I3Processor
{
 I5Processor()
 {
  System.out.print("Inside I5Processor() Constructor. ");
 }
}

public class JavaInheritance2
{
 public static void main(String[] args)
 {
  I5Processor i5 = new I5Processor();
 }
}
```

Inside I5Processor() Constructor. Inside I3Processor() Constructor. Inside Processor() Constructor.

Inside I5Processor() Constructor. Inside I5Processor() Constructor. Inside I5Processor() Constructor.

Inside Processor() Constructor. Inside I3Processor() Constructor. Inside I5Processor() Constructor.

Compiler error

Section 2 - Coding

Section Summary

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

Additional Instructions:

None

Q1. A company maintains a database that has the details of all the employees. There are two levels of employees where level 1 is the top management having a salary of more than 100 dollars and level 2 is the staff who are getting a salary of less than 100 dollars.

Create a class named Employee with empId and salary as attributes. Create another class empLevel that extends employee and categorizes the employee into various levels.

Input Format

The input should contain only the employee id and salary of the employee separated by space.
Employee id should be of integer type and salary float type.

Output Format

The output of the program must display the employee id, salary, and level of the employee one below the other in the same order.

Sample Input

```
253 5.6
```

Sample Output

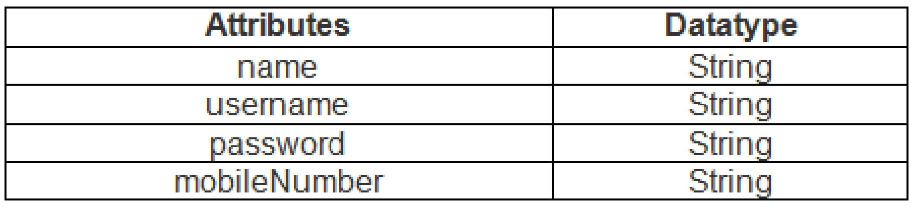
```
253
5.6
2
```

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

Q2. **User Matcher**

We are going to check duplicate user details. In our application, there can be only one user with a specific mobile number. If two users exist with the same mobile number they are duplicates. How to check whether two objects are equal?.Yeah, you are right the equals() method. Go on override the equals method and check if two users are same.

Create User class with following private attributes



Attributes	Datatype
name	String
username	String
password	String
mobileNumber	String

Include appropriate getters and setters.
Create Default and Parameterized Constructor as User(String name, String username, String password,String mobileNumber) for the class.
Override equals method that compares mobileNumber of the two objects.
Create a driver class named Main to test the above class.

Input Format

Input consists of two sets of user details
Refer sample input

Output Format

Output prints whether the two users are equal or not

Sample Input

```
Ankit
ank
ankit@26
9633310268
```

Sample Output

```
User 1 and User 2 are not equal
```

Sample Input

```
Ankit
ank
ankit@26
9633310268
```

Sample Output

```
User 1 and User 2 are equal
```

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

Answer Key & Solution

Section 1 - MCQ

Q1

Son’s Car

Solution

No Solution

Q2

2 3

Solution

No Solution

Q3

Using name of the interface.

Solution

No Solution

Q4

Won't compile because of line (5), constructor can't be static

Solution

No Solution

Q5

All classes above it

Solution

No Solution

Q6

It results in a compiler error

Solution

No Solution

Q7

A b = (String s) -> 1;

Solution

No Solution

Q8

option 4

Solution

No Solution

Q9 Inside Processor() Constructor. Inside I3Processor() Constructor. Inside I5Processor() Constructor.

Solution

No Solution

Section 2 - Coding

Q1 **Test Case**

Input

```
156 888
```

Output

```
156
888.0
1
```

Weightage - 20

Input

```
454 100
```

Output

```
454
100.0
2
```

Weightage - 20

Input

```
485 46
```

Output

```
485
46.0
2
```

Weightage - 10

Input

```
45 100.2
```

Output

```
45
100.2
1
```

Weightage - 20

Input

```
9658 88.000
```

Output

```
9658
88.0
2
```

Weightage - 20

Input

```
54 96
```

Output

```
54
96.0
2
```

Weightage - 10

Sample Input

```
253 5.6
```

Sample Output

```
253
5.6
2
```

Solution

```
 import java.util.Scanner;
class Employee
{
    public int empId;
    public float slry;

    public Employee(int empId, float slry)
    {
        this.empId = empId;
        this.slry = slry;
    }
    public String toString()
    {
        return (empId + "\n" + slry);
    }
}
class empLevel extends Employee
{
    public int level;
    public empLevel(int empId, float slry)
    {
        super(empId, slry);
        if (slry>100)
            level = 1;
        else
            level = 2;
    }
    public String toString()
    {
        return (super.toString() + "\n" + level);
    }
}
class empMain
{
    public static void main(String args[])
    {
        int emplId;
        float slry;
        Scanner in = new Scanner(System.in);
        emplId = in.nextInt();
        slry = in.nextFloat();
        empLevel el = new empLevel(emplId,slry);
        System.out.println(el.toString());
    }
}
```

Q2

Test Case

Input

```
david
sav
dav89
7890678990
```

Output

```
User 1 and User 2 are not equal
```

Weightage - 50

Input

```
neimer
neim
neimr!23
8907646789
```

Output

```
User 1 and User 2 are equal
```

Weightage - 50

Sample Input

```
Ankit
ank
ankit@26
9633310268
```

Sample Output

```
User 1 and User 2 are not equal
```

Sample Input

```
Ankit
ank
ankit@26
9633310268
```

Sample Output

```
User 1 and User 2 are equal
```

Solution

```
 import java.io.*;
import java.util.*;
class User {
 public User(String name, String username, String password, String mobileNumber) {
        this.name = name;
        this.username = username;
        this.password = password;
        this.mobileNumber = mobileNumber;
    }
 public User() {
     this.name = null;
     this.username = null;
     this.password = null;
     this.mobileNumber = null;
 }
private String name;
 private String username;
 private String password;
 private String mobileNumber;
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public String getMobileNumber() {
    return mobileNumber;
}
public void setMobileNumber(String mobileNumber) {
    this.mobileNumber = mobileNumber;
}
 public boolean equals(Object u) {
     if(u == this) {
         return true;
     }
     if (!(u instanceof User)) {
         return false;
     }
     User us = (User) u;
     return mobileNumber.compareTo(us.mobileNumber) == 0;
 }
}
class Main {
public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    User u1 = new User();
    User u2 = new User();
    u1.setName(sc.nextLine());
    u1.setUsername(sc.nextLine());
    u1.setPassword(sc.nextLine());
    u1.setMobileNumber(sc.nextLine());
    u2.setName(sc.nextLine());
    u2.setUsername(sc.nextLine());
    u2.setPassword(sc.nextLine());
    u2.setMobileNumber(sc.nextLine());
    if(u1.equals(u2)) {
        System.out.println("User 1 and User 2 are equal");
    }
    else {
        System.out.println("User 1 and User 2 are not equal");
    }

}
}
```