



Name:

Roll Number:

Section:

BSE-5B

3<sup>rd</sup> September, 2024

---

Q1. Find Error from the following code and correct it (if any) and write its EXACT output as it will be displayed on compiler

```
i.    public class InputAge {  
        public static void main(String[] args) {  
            Scanner input = new Scanner(System.in);  
            System.out.println("Enter your age: ");  
            int age = input.nextDouble();  
            int age = input.nextInt();  
            OR  
            Double age = input.nextDouble();  
  
            System.out.println("Your age is " + age);  
        }  
    }
```

Output here:

```

ii. public class User {
    private String name;
    private int age;
    public void setName(String newName) {
        name = newName;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int newAge) {
        age=newAge;
    }
}

class Main{
    public static void main(String[] args) {
        User obj = new User();
        obj.age = 25; //age is private so cannot be changes so create setter
        obj.setAge(25);
        System.out.print("Age: " + obj.getAge());
        System.out.println("Name: " + obj.getName); //brackets for function call
missing
        System.out.println("Name: " + obj.getName());
    }
}

```

Output here:

```

iii.    public class ChainingTest {
        public ChainingTest() {
            this(10);
            System.out.println("Default constructor");
        }
        public ChainingTest(int x) {
            System.out.println("Parameterized constructor: " + x);
        }
        public static void main(String[] args) {
            ChainingTest obj = new ChainingTest();
        }
    }

```

**Output here:**

**//no error in this one**

**Parameterized constructor: 10**

**Default constructor**

```

class Parent {
    private int x = 10;
    Parent() {
        System.out.println("Parent Constructor");
    }
}

class Child extends Parent { //final class cannot extend also Child class cannot be static
    int y = 20;
    Child() {
        super(); //super() must be the 1st statement in child's constructor if used
        System.out.println("Child Constructor");
        this(); //this will call infinite chaining for child constructor
        x=100; //cannot access private variable of parent class
    }
    void display() {
        System.out.println("Child display: " + this.y);
        System.out.println("Parent display: " + super.x);
    }
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.display(); //parent object cannot access child's function
    }
}

```

**Output here:**

**Q2. Write java code to take two strings from user using JOptionPane. Display the frequency of second string in first string. Also display starting index of occurrences. Suppose:**

**String1 = abcdabcdeabc**

**String2= abc**

**Frequency = 3**

**Indexes = [0, 4, 9]**

```
import javax.swing.JOptionPane;
```

```
import java.util.ArrayList;
```

```
public class StringFrequency {
```

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

```
        // Taking input from user using JOptionPane
```

```
        String str1 = JOptionPane.showInputDialog("Enter the first string:");
```

```
        String str2 = JOptionPane.showInputDialog("Enter the second string:");
```

```
        // Find frequency and indexes of occurrences
```

```
        int frequency = 0;
```

```
        ArrayList<Integer> indexes = new ArrayList<>();
```

```
        int index = str1.indexOf(str2);
```

```
        while (index != -1) {
```

```
            frequency++;
```

```
            indexes.add(index);
```

```
            index = str1.indexOf(str2, index + 1);
```

```
        }
```

```
// Displaying results  
JOptionPane.showMessageDialog(null, "Frequency = " + frequency + "\nIndexes = " +  
indexes);  
}  
}
```