COMP1020 Winter 2024

In-Class Participation Week 4 Tuesday

Q1) What is the output? **[1M]**

```
import java.util.*;


public class Q1 {
   public static void main(String args[]) {
      ArrayList<String> myArrList = new ArrayList<String>();
      String p1 = new String("John");
      String p2 = new String("Fred");
      myArrList.add(p1);
      myArrList.add(p2);
      myArrList.add(p2);
      myArrList.add(p1);
      myArrList.add(p1);
      myArrList.add(p2);
      myArrList.add(p1);
      myArrList.add(p2);
      myArrList.add(p1);


      System.out.println(myArrList.contains(new String("John")));
      System.out.println(myArrList.contains(p1));
      System.out.println(myArrList.indexOf(new String("Jane")));
      System.out.println(myArrList.indexOf(p2));
      System.out.println(myArrList.lastIndexOf(new String("Fred")));
      System.out.println(myArrList.lastIndexOf(p2));
   }
}
```


Q2) What is the output? **[1M]**

```
import java.util.*;


public class Q2 {
   public static void main(String args[]) {
      ArrayList<String> myArrList = new ArrayList<String>();
      String sb1 = new String("Jan");
      String sb2 = new String("Feb");
      myArrList.add(sb1);
      myArrList.add(sb2);
      myArrList.add(sb2);
      ArrayList<String> assignedArrList = myArrList;
      ArrayList<String> clonedArrList = (ArrayList<String>) myArrList.clone();
      System.out.println(myArrList == assignedArrList);
      System.out.println(myArrList == clonedArrList);
      String myArrVal = myArrList.get(0);
      String assignedArrVal = assignedArrList.get(0);
      String clonedArrVal = clonedArrList.get(0);
      System.out.println(myArrVal == assignedArrVal);
      System.out.println(myArrVal == clonedArrVal);
   }
}
```


Q3) You are working on a project that involves managing Employees for HR Dept. Create a Java a program that demonstrates **deep and shallow copies** of employee objects.

Define an **Employee** class with the following attributes**:** **[2M]**

- **id (int)**
- **name (String)**
- **salary (float)**
- **department (String)**
- Implement the **clone()** method that creates a new Employee object.
- **toString()** to print the details of the Employee object.
- you can also add getters and setters.

EmployeeTest class - main():

* Create an array of Employee objects called **empDataHR .**

* The HR department wants to create **empDataHRtemp** array which point to the same data present in **empDataHR**.

* The payroll dept requests the same employee data. Call this array of objects as **empDataPayroll** and deep copy the **empDataHR**.

* Modify the original and cloned objects with the help of setters to demonstrate the differences between deep and shallow cloning by printing the output.

Q4) Bonus Question **[2M]**

Question-related to slides number: 93-95

(No teamwork allowed. Students should be able to explain all the questions asked related to the code.)

