Advanced Programming (CSE201), Midsem Quiz
Time allocated: 03:00pm – 4:00pm (1 hour), Total Marks: 20
(No Submissions are allowed beyond 4.10pm)

Instructions:

- This is a closed book quiz.
- Only reasonable and clearly mentioned assumptions (if any) would be accepted.
- For justifications, please be as concise as possible (2-3 sentences only)
- **IIIT plagiarism policy is applicable if any such cases found**
- **Write your answers on a plain sheet that you can upload by taking a picture of the same (ensure low resolution so that the upload size is smaller)**
- **You can either submit your solutions on the below Google Form link https://docs.google.com/forms/d/e/1FAIpQLSe6yba9jd_k5KnHt071_9fTYj8VJWAnSgyHcxHUE21N2aAPIw/viewform?usp=sf_link or email the quiz solutions to "ap-m2020-submission@iiitd.ac.in" or upload on google classroom . Only one mode of submission!**
- **Subject of the mail should be the Midsem Quiz.**
- **We will not consider any submission that is submitted beyond 4.10pm. It is your responsibility to ensure you email it or submit through google form or through google classroom on time. Please ensure you have proper internet connectivity as we are giving you sufficient extra time to send the email or upload your solutions either on google classroom or on google form.**

Q1) Explain why Java does not support multiple inheritance by taking the reference of the diamond problem. Feel free to use the resources on the Internet for this question, but include appropriate citation(s) in your answer. You must write your answer in your own words instead of copy-pasting from the Internet. **[2 marks]**

Q2) The general intent of cloning is to satisfy the following three soft requirements.

- x.clone().equals(x)
- x.clone() != x
- x.clone().getClass() == x.getClass()

a) In the following code fragment, indicate the lines that satisfy each of the above three requirements. In addition to that, also explain why/how they satisfy the three requirements? **[6 marks]**

```
1. public Point3D clone() {
2.     Point3D d = (Point3D) super.clone();
3.     d.z = this.z;
4.     return d;
5. }
```

b) Does the below clone() method violate any of the three requirements mentioned above? Explain your answer. **[2 marks]**

```
1.  public class Institute implements Cloneable {
2.     private String instName;
3.     private List<String> employeeNames;

4.  @Override
5.  public Institute clone() {
6.        try {
7.           Institute copy = (Institute) super.clone();
8.           return copy;
9.        } catch (CloneNotSupportedException e) {
10.          e.printStackTrace();
11.          return null;
12.       }
13.    }
14. }
```

Q3) Rewrite the following class so that it becomes immutable. Do not assign values to the instance variables directly at the time of declaration. **[2 marks]**

```
1. public class MakeThisImmutable {

2.       public int id;
3.       protected String name;
4.       public Student stud;

5.       public int getID() {
6.                return id;
7.       }
8.       public void setID(int id) {
9.                this.id = id;
10.      }
11.      public String getName() {
12.               return name;
13.      }
14.      public void setName(String name) {
15.               this.name = name;
16.      }
17.      public Student getStudent() {
18.               return new Student();
19.      }
20.      public void setStudent(Student stud) {
21.               this.stud = stud;
22.      }
23. }
```

```
1. public class Student {
2.       private String name;

3.       public Student() {
4.                name = "noname";
5.       }
6.       public String getName() {
7.                return name;
8.       }
9.       public void setName(String name) {
10.               this.name = name;
11.      }
12. }
```

Q4) Construct the skeleton code that could lead to the following stacktrace. We expect the exact class and method names. Include the line numbers wherever appropriate. **Do not use loops**. Handle the exceptions, i.e there should be try and catch blocks. Using your skeleton code, explain how it would end up with each of the three exceptions listed here. **[4 marks]**

```
1. java.lang.NullPointerException
2.         at University.getCourses(University.java:25)
3.         at University.enrollIntoCourse(University.java:15)
4.         at University.teachCourse(University.java:7)
5.         at University.main(University.java:35)
6.  java.lang.NullPointerException
7.         at University.getCourses(University.java:25)
8.         at University.enrollIntoCourse(University.java:15)
9.         at University.teachCourse(University.java:7)
10.        at University.main(University.java:35)
11.  java.lang.NullPointerException
12.        at University.getCourses(University.java:25)
13.        at University.enrollIntoCourse(University.java:15)
14.        at University.teachCourse(University.java:7)
15.        at University.main(University.java:35)
```

Q5) Consider the code given below. **[4 marks]**

a) Write the output of the two print statements.
b) What is the value of the instance variables after deserialization?
c) In case this code results in an error, explain the reason.

```
1. public class Organization implements Serializable {

2.          private static final long serialVersionUID = 20L;
3.          private int orgID;
4.          private transient List<String> employeeNames;
5.          private transient double profit;


6.          public Organization(int orgID, List<String> employeeNames, double profit) {
7.                    this.orgID = orgID;
8.                    this.employeeNames = employeeNames;
9.                    this.profit = profit;
10.         }

11.         public void serialize() throws FileNotFoundException, IOException {
12.                   ObjectOutputStream out = null;
13.                   try {
14.                             out = new ObjectOutputStream(new FileOutputStream("out.txt"));
15.                             out.writeObject(this);
16.                   } finally {
17.                             out.close();
18.                   }
19.         }

20.         public Organization deserialize() throws FileNotFoundException, IOException, ClassNotFoundException {
21.                   ObjectInputStream in = null;
22.                   Organization org = null;
23.                   try {
24.                             in =  new ObjectInputStream (new FileInputStream("out.txt"));
25.                             org = (Organization) in.readObject();
26.                   } finally {
27.                             in.close();
28.                   }
29.                   return org;
30.         }

31.         @Override
32.         public String toString() {
33.                   return orgID + " " + employeeNames.toString() + " " + profit;
34.         }

35. }
```

```
1.  public static void main(String[] args) {
2.                  List<String> empList = new ArrayList<String>();
3.                  empList.add("John");
4.                  empList.add("Mike");
5.                  Organization org = new Organization(10, empList, 100.02);
6.                  System.out.println(org);
7.                  try {
8.                          org.serialize();
9.                          Organization deserializedOrg = org.deserialize();
10.                         System.out.println(deserializedOrg);
11.                 } catch (Exception e) {
12.                         e.printStackTrace();
13.                 }
14.       }
```