CMSC 132 Quiz 1 Worksheet

The first Quiz of the course will be on Wednesday, Sept 7 during your lab (discussion) session. The following list provides more information about the quiz:

- You will have 25 minutes to complete the quiz.
- It will be a written quiz (not using any computer).
- It will be closed-book, closed-notes, and no calculator is allowed.
- Answers must be neat and legible. We recommend that you use pencil and eraser.
- The quiz will be based on the exercises you will find below.

The following exercises cover the material to be included in Quiz #1. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with TAs and instructors during office hours.

**Problem 1**

1. Consider the following two classes, **Oven** and **Microwave**:

```java
public class Oven {
    protected String make;

    public void turnOn() {
        System.out.println("Oven On");
    }
}
```

```java
public class Microwave extends Oven {
    public double power;

    public void turnOn() {
        System.out.println("Micro On");
    }

    public void turnOn(int thePower) {
        power = thePower;
        System.out.println("Micro On with " + power);
    }
}
```

Based on these classes, answer the following questions:
a. The following main is in a class named Driver. The Driver, Oven and Microwave classes all belong to the same package. Indicate whether each of the underlined statements below is valid or invalid. (Consider each one in isolation; earlier errors do not affect later statements.)

```java
public static void main(String[] args) {
    Oven p = new Oven(); // These two variable declarations are correct
    Oven m = new Microwave();

    p.turnOn(10);  // p.make = "GT";
    p.make = "GT";  // p.power = 200;
    p.power = 200;
    m.turnOn(20);
    m.turnOn(20);
    Microwave k = (Microwave)m;
    k.turnOn(200);
    Microwave.make = "TU";
    k.turnOn(200);
    Microwave.make = "TU";
    Oven.make = "TU";
    Oven.make = "TU";
    Oven.turnOn();
}
```

b. The following statements are also part of main. Show the output produced by each. If an error results, write “ERROR”.

```java
Oven v1 = new Microwave();
v1.turnOn();
```

Output:

```
```

```java
Microwave v2 = new Oven();
v2.turnOn();
```

Output:

```
ERROR
```
Problem 2
Consider the interface Wireless and class Network below.

```java
public interface Wireless {
    public double getRange();
}
```

```java
public class Network {
    private String name;
    private int numComputers;

    public Network(String n, int c) {
        name = n;
        numComputers = c;
    }

    public Network(Network n) {
        name = new String(n.name);
        numComputers = n.numComputers;
    }

    public String getName() {
        return name;
    }

    public int getNumComputers() {
        return numComputers;
    }
}
```
Implement a class named **WirelessNetwork**, that extends the class **Network** and implements the **Wireless** interface. It has a single private double instance variable, called **maxRange**, which represents the wireless network’s maximum signal range. Your class should contain the following public methods:

a. **Constructor**: This constructor takes three arguments. A String parameter representing the network name, a number of computers, and a double representing the maximum range.

b. **Default Constructor**: The constructor will initialize a WirelessNetwork object with the values “COMP_NAME”, 1, and 50. **Note**: You must use the constructor from part a. in your implementation of this constructor.

c. **equals**: A boolean method `equals(Object o)`, which tests this object for equality with another object. To be equal, both objects must be of the same class (recall the method `getClass`) and have the same name, number of computers and maximum range.

d. **Interface method(s)**: implement any method(s) necessary to satisfy the **Wireless** interface. The wireless device range is just the value of maxRange.

**Problem 3**

Write a method called **clean** that has the following prototype:

```java
public static String[][] clean(String[][] data, int number);
```

The method returns a two-dimensional array with those rows of **data** whose length is greater than or equal to number.