- Look at the java code and answer the following questions.

1- How many user-defined methods are defined in the code? Looking at their headers (signatures), can you specify what inputs and outputs they have? Do you see method overloading?

4 user-defined methods
public static int foo(int num)
input: int
output: int

public static int max (int a, int b)
input: 2 int
output: int

public static String foo(double num)
input: double
output: String

public static void displayCurrentUser()
input: none
output: none

foo method is overloaded!

2- Can you write the output of this program? (For current time method, you don't need to calculate it, just write the current time)

Before method call
Inside method foo. num is 13
return value is: 14
Inside method foo. num is 154
a: 56
x: 14 y: 0 z: BIGNUM
Current time is 1:52:6 GMT
After method call
public class Methods {

    public static int foo(int num) {
        System.out.println("Inside method foo. num is "+ num);

        if (num % 2 == 1) { //odd number
            num++;
        } else {
            return 0;
        }

        System.out.println("return value is: "+ num);
        return num;
    }

    public static int max (int a, int b) {
        if (a > b)
            return a;
        else
            return b;
    }

    public static String foo(double num) {
        if (num > 20) {
            return "BIGNUM";
        } else {
            return "littlenum";
        }
    }

    public static void displayCurrentTime() {

        // Obtain the total milliseconds since the midnight, Jan 1, 1970
        long totalMilliseconds = System.currentTimeMillis();

        // Obtain the total seconds since the midnight, Jan 1, 1970
        long totalSeconds = totalMilliseconds / 1000;

        // Compute the current second in the minute in the hour
        int currentSecond = (int)(totalSeconds % 60);

        // Obtain the total minutes
        long totalMinutes = totalSeconds / 60;

        // Compute the current minute in the hour
        int currentMinute = (int)(totalMinutes % 60);

        // Obtain the total hours
        long totalHours = totalMinutes / 60;

        // Compute the current hour
        int currentHour = (int)(totalHours % 24);

        // Display results
        System.out.println("Current time is "+ currentHour + ":"+ currentMinute + ":"+ currentSecond + " GMT");
    }
}
```java
public static void main(String[] args) {
    System.out.println("Before method call");
    int x = foo(13);
    int y = foo(154);
    String z = foo(154.0);
    int a = max(24, 56);
    System.out.println("a: " + a);
    System.out.println("x: " + x + " y: " + y + " z: " + z);
    displayCurrentTime();
    System.out.println("After method call");
}
```