

**CS212**  
**Midterm**

1. Read the following code fragments and answer the questions.

(a)

```
public void displayAbsX(int x) {
    if (x > 0) {
        System.out.println(x);
        return;
    }
    else {
        System.out.println(-x);
        return;
    }
    System.out.println("Done");
}
```

What is the flaw that stops the above method from compiling?

**Answer:** The last line of the code cannot be reached since both the if and else clauses return from the method.

(b)

```
class X {
    public int meth(int a, int b) { ... }
    public String meth(int a, int b) { ... }
}
```

Does the above code fragment compile? Explain.

**Answer:** These two overloaded methods use identical parameters. This is illegal. Therefore, the program will not compile.

(c)

```
public boolean hasDuplicateValues(int[] data) {
    for(int i = 0; i < data.length; i++)
        for(int j = 0; j < data.length; j++)
            if(data[i] == data[j]) return true;
    return false;
}
```

Does the above method work properly? If it doesn't, fix it.

**Answer:** The code fragment will give an incorrect result if there are no duplicate values in the data. Since `data[0] == data[0]`, the first comparison already returns true!. This error can be remedied by replacing `j = 0` by `j = i + 1`. This way, only actual duplicates will be found.

(d) 

```
String names = { "George", "susan" } ;
int totalLength = 0;
for (int i = 0; i < names.length(); i++)
    totalLength += names[i].length;
```

Find the error in the above code fragment.

**Answer:** This code fragment has three errors. The first is the missing square brackets in the array declaration.

```
String[] names = {"George", "Susan"}; //corrected statement
```

The second error: the length of the array is given by `names.length`. It's not a method. So, it doesn't have attached empty parentheses.

The third error: `totalLength` is the sum of the length of the two String elements of the array `names`.

The length here is a method. So it should be written as `names[i].length()`;

(e) 

```
public class Oops {
    private int x = 3;

    public static void changeX() {
        x = 4;
    }
}
```

What is the error in the above code fragment?

**Answer:** An instance variable can't be accessed from a static method in the same class without an object and an accessor method. The method `changeX()` shouldn't be a static method.

2. Please complete each part before moving to the next.

- (a) Create a class `Android` whose objects have unique names. The class has the following attributes:
- `tag` - a static integer that begins at 1 and changes each time an instance is created.
  - `name` - a string that unique for each instance of this class.
- Create the class and declare the instance variables.

```
public class Android {
    private static int tag = 1;
    private String name;

}
```

- (b) Write a default constructor that changes the *tag* to the next prime number by calling the private method **changeTag()**, and then sets the name to "Bob" concatenated with the value of *tag*.

**Answer:**

```
public Android() {  
    changeTag();  
    name = "Bob" + tag;  
}
```

- (c) Write a **getName()** method that returns the name of the Android object.

```
public String getName(){  
    return name;  
}
```

- (d) Write a private static method **isPrime(n)** that returns true if n is prime - that is, if it is not divisible by any number from 2 to n-1.

```
private static boolean isPrime(int n){  
    for(int i = 2; i < n; i++)  
        if(n % i == 0) return false;  
    return true;  
}
```

- (e) Write a private static method **changeTag()** that replaces *tag* with the next prime number larger than the current value of *tag*.

```
private static int changeTag() {  
    tag++;  
    if(!isPrime(tag)) tag++;  
    return tag;  
}
```

3. Here is a code for a recursive method **mystery**. What is printed when **mystery(2,2)** is called?

```
public class TestMystery {  
  
    public static void main(String[] args) {  
        mystery(2,2);  
    }  
  
    public static void mystery(int a, int b) {  
        if( a == 0 && b == 0)  
            System.out.println(0);  
        else if ( a == 0) {  
            //System.out.println(a, b-1); //This line has error  
            System.out.println(b); //This is the correct statement  
            mystery(a, b-1);  
        }  
        else {  
            mystery(a-1, b);  
            System.out.println(b);  
        }  
    }  
}
```

Write the output in the order it will be printed when the program is run on the computer.

The output from the program is:

2  
1  
0  
2  
2

When the method is first called, it enters the else branch, where the function is recursively called each time with a smaller value of a. However, output doesn't get printed here because the print statement is after the recursive call.

When a becomes 0, then program enters the else if block and prints b before recursively calling the function with the smaller value of b.

The first output is from this block followed by the output from the if block when both a and b are zero and finally the output from the else block is printed.

Note: Every one was given 4 points for this question due to the error in the question.

4. A palindrome is any word, phrase, or sentence that reads the same forward and backward. Here are some well-known palindromes:

Able was I ere I saw Elba  
Desserts I stressed  
kayak

Write a complete java program that uses recursion to determine whether a string argument is a palindrome. The program should have a main method and a static recursive isPalindrome method.

```
import java.util.Scanner;

public class Palindrome {
    public static void main(String[] args){
        String str;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a string to test if it is a palindrome: ");
        str = input.nextLine();
        //At this point, if there are commas etc in the text one can remove
        // them at this time. Even spaces can be removed.
        str = str.toLowerCase();
        if (isPalindrome(str))
            System.out.println("The string, " + str + ", is a palindrome");
        else
            System.out.println("The string, " + str + ", is not a palindrome");
        input.close();
    }

    public static boolean isPalindrome(String s) {
        int len = s.length();
        if (len <= 1) return true;
        return (s.charAt(0) == s.charAt(len-1)) &&
            isPalindrome(s.substring(1, (len-1)));
    }
}
```