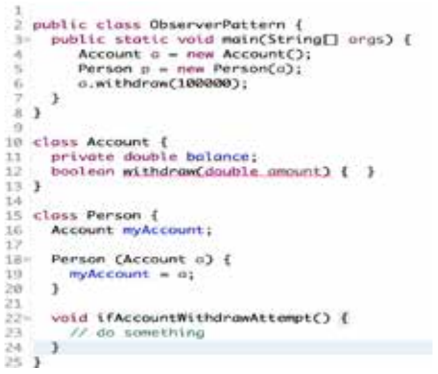
COMS 309 Week#13 Mon Worksheet-1

TEAM#? _______________

NET-IDs of students who are present: ____

1 Design



```java

public class ObserverPattern {
  public static void main(String[] args) {
     Account a = new Account();
     Person p = new Person(a);
     a.withdraw(100000);
  }
}

class Account {
  private double balance;
  boolean withdraw(double amount) {  }
}

class Person {
  Account myAccount;

  Person (Account a) {
    myAccount = a;
  }

  void ifAccountWithdrawAttempt() {
    // do something
  }
}
```

Rewrite above code to use the Observer Pattern. Show ALL the changes needed. *SHOW YOUR NEW CODE.*

The Account is the subject and the Person is the observer and would like to be notified if withdraw is being called.

Step 1: Make the Account class extend Observable and change the withdraw method.

```
class Account extends Observable {
   …
   boolean withdraw {
      …
      setChanged();
      notifyObservers(); //with or without data
   }
}
```

Step 2: Make the Person class implements Observer and update method

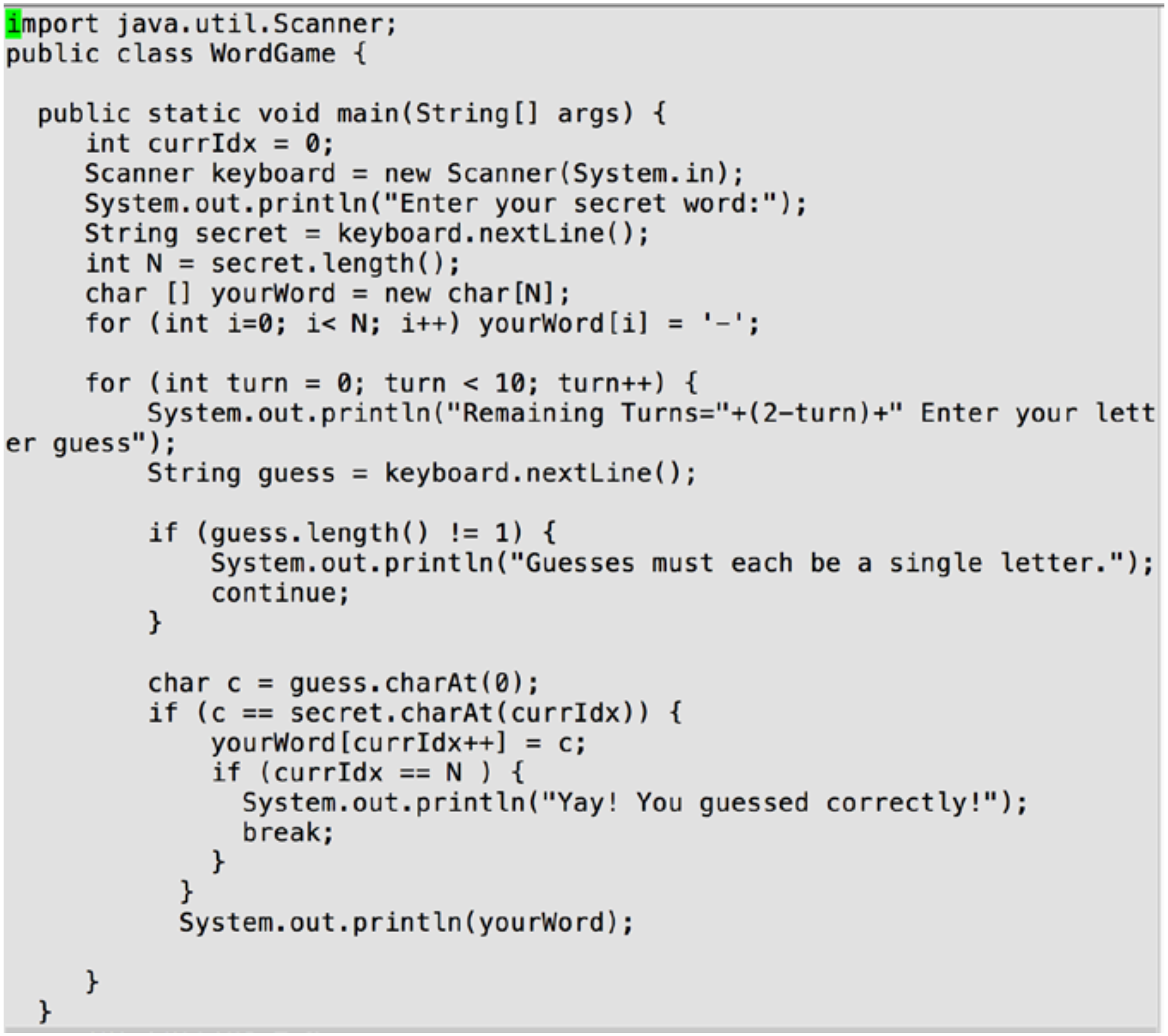
```
class Person implements Observer {
   …
   void update (Observable o, Object arg) {
      // this is called when withdraw is made
      // may use arg if provided.
   }
}
```

Step 3: In the Application (ObserverPattern class here), add p as an observer to the account a.

```
Account a = new Account();
Person p = new Person(a);
a.addObserver(p);
a.withdraw(100000); // will notify person p
```

Note that in this specific case, Person p does know about his/her account – and that's ok. In general, Observer may not know about the Subject.

COMS 309 Week#13 Mon Worksheet-1



```
import java.util.Scanner;
public class WordGame {

  public static void main(String[] args) {
     int currIdx = 0;
     Scanner keyboard = new Scanner(System.in);
     System.out.println("Enter your secret word:");
     String secret = keyboard.nextLine();
     int N = secret.length();
     char [] yourWord = new char[N];
     for (int i=0; i< N; i++) yourWord[i] = '-';

     for (int turn = 0; turn < 10; turn++) {
         System.out.println("Remaining Turns="+(2-turn)+" Enter your lett
er guess");
         String guess = keyboard.nextLine();

         if (guess.length() != 1) {
             System.out.println("Guesses must each be a single letter.");
             continue;
         }

         char c = guess.charAt(0);
         if (c == secret.charAt(currIdx)) {
             yourWord[currIdx++] = c;
             if (currIdx == N ) {
               System.out.println("Yay! You guessed correctly!");
               break;
             }
           }
           System.out.println(yourWord);

     }
  }
```

Assume that the above code will be re-written using MVC pattern.

Write down method signatures of the Model for the word search game.

Write down method signatures for the View for the word search game.

Model Responsibilities:
→ Store data about game (like turns, word)
→ provide game methods

void setSecretWord(String s);

int attemptGuess(char c);
// return value will indicate success

boolean hasMoreTurns();

String getGameState();

View Responsibilities:
→ draw/show game state/information

void displayGameState(String s);

void displayMessage(String s);

2 Testing

Q:

```
String guess = keyboard.nextLine();

if (guess.length() != 1) {
    System.out.println("Guesses must each be a single letter.");
}
```

a) Write test cases that will achieve 100% statement coverage.

1. Abc // for this input, all statements will be executed.

b) Write test cases that will achieve 100% branch coverage.

1. Abc // only TRUE statement of if would be executed
2. A // also FALSE will be executed.

Q: A switch is switched on when the temperature falls below 50 degrees and then turned off when the temperature is above 70 degrees. Identify the equivalence classes for testing the switch.

Class 1: temperature < 50
Class 2: temperature >= 50 and <= 70
Class 3: temperature > 70

Q: Given code for SORTING (say ascending), write down

a) Signature of Oracle

boolean isSorted (int[] input, int[] result)

What is an oracle?

There are two SEPARATE things. The SOLVER (which is also SUT or system under test) and the CHECKER (which is the oracle). The solver is given input and gives the result. The checker takes the given input AND the result and then CHECKS if the results are correct. If so, it returns TRUE. Otherwise, it returns FALSE.

b) Logic of Oracle

1) Make sure all elements of input are in result (and with right frequency too).
2) Make sure that the result is sorted (i.e. elements are in ascending order).

Q: What is regression testing? Explain why it is important to do regression testing.

See Mock1

3 Ethics

Assume that you are currently designing a database management system for the personnel office of ABC a mid-sized company. You have been involving the management in the design process from the start of the project. Now it's time to decide about the kind of degree of security to build into the system. You described several options to the client. The client/ ABC Management decided to opt for the least secure system because the system is going to cost more than was initially planned, and the least secure option is the cheapest option. You know that the DB includes sensitive information, you feel that the weak security would make the system vulnerable, but the management is convinced that the cheapest security is what they want.

Q. What is one way to convince everyone that this is an ethical issue worthy of being further examined?

Smell Test. See Mock1.

Q. Describe application of at least THREE of the five ethical approaches to the above problem.

I will let you work on this!

a) Approach 1:
b) Approach 2:
c) Approach 3: