TEAM#? _____

NET-IDs of students who are present: _____

# 1 Design

```
2  public class TestClass {
3
4    public static void main(String[] args) {
5       // This is the CLIENT code.
6      HighLevel l = new HighLevel();
7      l.doOperation();
8    }
9
10 }
11
12 class HighLevel {
13
14   MyDatabase d;
15
16   void HighLevel() {
17     MyDatabase d = new MyDatabase();
18   }
19
20   void doOperation() {
21     // do lots of things
22     // then use table (say)
23     d.useTable();
24   }
25 }
26
27 class MyDatabase {
28   void useTable() {}
29 }
```

**Rewrite code to INJECT database dependency into the HighLevel module (from the main method). Show ALL the changes needed. *SHOW YOUR NEW CODE.***

```
public class Game {

 // check if args[2] is an int
 public static boolean CheckInt(String input) {
  boolean valid = true;
  for (int i = 0; i < input.length(); i++) {
   String secondarg = input.substring(i, i + 1);
   if (secondarg.compareTo("0") < 0 || secondarg.compareTo("9") > 0) {
    valid = false;
    System.out.println("First argument must be integer");
    System.exit(1);
   }

  }
  return valid;
 }

 public static void main(String[] args) {
  BufferedReader reader = null;
  if (args.length != 2) {
   System.out.println("Please enter two arguments");
   System.exit(2);
  }

  boolean result = CheckInt(args[1]);

  // try to open file
  try {
   reader = new BufferedReader(new FileReader(args[0]));
  } catch (FileNotFoundException fnfe) {
   System.out.println("Error opening file" + args[0]);
   System.exit(3);
  }
  boolean done = false;
  String inputLine = null;
  String[] words = null;
  int length = 0;

  while (!done) {
   try {
    inputLine = reader.readLine();
   } catch (IOException ioe) {
    System.out.println("I/O error");
    System.exit(4);
   }

   //end of file
   if (inputLine == null) {
    done = true;
   } else {
    String line = inputLine;
    String delimter = " ";
    words = line.split(delimter);
    length = words.length;
   }
  }

  //random number generator to select word from array
  Random generator = new Random();
  int num = generator.nextInt(length - 0) + 0;

  //store chosen word
  String chosen = words[num];

  //convert string to char array
  char[] chosenarr = chosen.toCharArray();
  char[] output = new char[chosenarr.length];

  for (int x = 0; x < output.length; x++) {
   output[x] = '*';
  }
  System.out.println(new String(output));
```

```
  //user guesses
  int count = 0;
  int guessnum = Integer.parseInt(args[1]);

  BufferedReader userguess = new BufferedReader(new InputStreamReader(System.in));
  String guess = null;
  char[] userlinechar = null;
  String userlinestring = null;
  String b = null;
  String c = null;

  for (int a = 0; a < guessnum; a++) {

   System.out.println("Guess a character..");

   try {

    //convert bufferedreader to chararray
    guess = userguess.readLine();
    userlinechar = guess.toCharArray();
    userlinestring = Character.toString(userlinechar[0]);

    //check if user guess is valid
    if (!userlinestring.matches("[A-Za-z]+") || userlinechar.length != 1) {
     System.out.println("Enter valid input");
     a--;
    }

    //compare user guess with chosen word
    for (int x = 0; x < chosenarr.length; x++) {


     for (int j = 0; j < output.length; j++) {
      c = new String(chosenarr);
      b = new String (output);

     if (Character.toLowerCase(userlinechar[0]) == Character.toLowerCase(chosenarr[x])) {
      output[x] = Character.toLowerCase(userlinechar[0]);

     }

     }

     }

      if (b.equalsIgnoreCase(c)){
     break;
     }

   } catch (IOException ioe) {
    System.exit(4);
   }
  }

  System.out.println(new String(output));

  // check if user has won
  int countLose = 0;

  for (int i = 0; i < output.length; i++) {
   if (output[i] == '*') {
    countLose++;
   }
  }

  if (countLose > 0) {
   System.out.println("Hard luck");
  } else {
   System.out.println("Well done");
  }
 }
}
```

Redo above using MVC OR MVP. Show CODE for your classes.

## 2 Testing

```java
public int binarySearch(int[] inputArr, int key) {

    int start = 0;
    int end = inputArr.length - 1;
    while (start <= end) {
        int mid = (start + end) / 2;
        if (key == inputArr[mid]) {
            return mid;
        }
        if (key < inputArr[mid]) {
            end = mid - 1;
        } else {
            start = mid + 1;
        }
    }
    return -1;
}
```

Q: Consider the above code. Generate test cases for it using the following techniques (first write a brief description of the technique):

a) Boundary-value testing
b) Equivalence-class testing
c) Statement coverage
d) Decision coverage

Q: For each step in testing, write down how it can be replaced/automated:

a) Generating test cases.
b) Figuring out expected results.
c) Writing code for test cases.
d) Running test cases.

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

Q: What is a driver? What is a stub? What are mock objects?

Q: How can you drive Web Server testing?

Q: How can you mock Web Servers in order to test Web Clients?

Q: How can you drive Android UI testing?

# 3 Ethics

Q. What is a way to check if an issue/decision requires an ethical treatment?

Q. What are the five approaches to an Ethical Decision? Use an example to explain.

Q. What are the eight principles of IEEE's code of conduct?