**Please provide your answers in the space to the right:**

1. **(6 pts)** What is the output of the following program:

***public class MyClass1 {***

***protected int x = 1;***

***public void display1() {***

***System.out.println(“x = “ + x);***

***}***

***}***

***public class MyClass2 extends MyClass1 {***

***MyClass2() {***

***x = 2;***

***}***

***}***

***public class MyClass3 extends MyClass2 {***

***MyClass3() {***

***x = 3;***

***}***

***public void display3() {***

***System.out.println(“x = “ + x);***

***}***

***}***

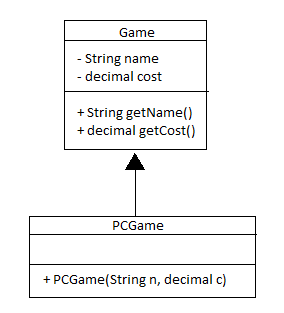
***public static void main(String[] args) {***

***MyClass3 c3 = new MyClass3();***

***c3.display1();***

***c3.display3();***

***}***

1. **(10 pts)** In the space to the right, implement the following classes. Assume that the PCGame constructor assigns values to the *name* and *cost* fields using the arguments it is passed.

**Answers**

1)

2)

1. In the space to the right, initialize a multi-dimensional int array that takes the following form and has these values:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |

1. **(5 pts)** using nested ***for*** loops
2. **(5 pts)** using ***array[i] = { a, b, c, … }*** notation.
3. **(5 pts)** using ***array = { a, b, …, z }*** notation.
4. **(12 pts)** In the space to the right, create an ***enum*** called ***TileType*** that contains these instances: ***FLOOR, WALL\_NS, WALL\_WE,*** and ***CORNER***. Write a class called ***Tile*** that contains the following fields: ***int posX, int posY, TileType type***. Add a constructor for this class that accepts arguments to initialize each of the three fields. In the main method below create four instances of the ***Tile*** class, one for each of the ***TileType*** enum constants.

***public class EntryPoint {***

***public static void main(String[] args) {***

***Tile floor = new Tile(FLOOR);***

***Tile wall\_NS = new Tile(WALL\_NS);***

***Tile wall\_NS = new Tile(WALL\_NS);***

***Tile wall\_NS = new Tile(WALL\_NS);***

***}***

***}***

1. **(5 pts)** Find the errors in the following interface declaration. Put a corrected version of this interface in the space to the right.

interface Physics {

public static float GRAVITY = 9.8f;

private static float velX = 0.0f;

private float velY = 0.0f;

private float CalculatePosX();

float CalculatePosY();

}

3a) ***int[][] ar = new int[4][3];***

***for(int i = 0; i < 4; i++) {***

***for(int j = 0; j < 3; j++) {***

***ar[i][j] = i \* 3 + j;***

***}***

***}***

3b) ***int[][] ar = new int[4][3];***

***ar[0] = { 1, 2, 3 };***

***ar[1] = { 4, 5, 6 };***

***ar[2] = { 7, 8, 9 };***

***ar[3] = { 10, 11, 12 };***

3c) ***int[][] ar = new int[4][3];***

***ar = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };***

4) ***public enum TileType {***

***FLOOR,***

***WALL\_NS,***

***WALL\_WE,***

***CORNER;***

***}***

***public class Tile {***

***int posX, posY;***

***TileType type;***

***public Tile(int x, int y, TileType type) {***

***posX = x;***

***posY = y;***

***this.type = type;***

***}***

***}***

5) ***Can’t have private fields***

***Need static fields***

***Can’t have private methods***

***interface Physics {***

***public static float GRAVITY = 9.8f;***

***public static float velX = 0.0f;***

***public static float velY = 0.0f;***

***public abstract float CalculatePosX();***

***public abstract float CalculatePosY();***

***}***

1. **(15 pts)** Write a program that reads in words from the console and stores them in an **ArrayList**. Each word is entered on a new line on the console, and the last word is always “***END***”. The program should display all of the words that have been entered and a count of the total number of words read by the program.

**import java.util.\*;**

**public class Program {**

**public static void main(String[] args) {**

**/\* your**

**code**

**goes**

**here \*/**

**}**

**}**

1. **(15 pts)** Complete the following program:
   1. Write another program that uses a ***BufferedReader*** to read text from a file that is passed in as a command line argument. The text should be printed to the console. (**6 pts)**
   2. Make sure that you handle all possible exceptions that occur. (**6 pts)**
   3. Print an error message for each error that you catch to help the user properly run your program. (**6 pts)**

**import java.io.\*;**

**public class Program {**

**public static void main(String[] args) {**

**/\* your**

**code**

**goes**

**here \*/**

}

}

1. **(12 pts)** Using the diagram on the next page, list all methods available to the Avatar class. For any parent methods not available to the Avatar class, list 3 ways that you could make them available and concerns you may have in implementing each.

6***)*   *Scanner kb = new Scanner(System.in)***

***ArrayList al = new ArrayList();***

***int wordCount = 0;***

***String currentWord = “”;***

***while(true) {***

***currentWord = kb.next();***

***if(currentWord.equals(“END”)***

***break;***

***else al.add(currentWord);***

***}***

***System.out.print(“You entered “+al.size()+“words:”);***

***for(int i = 0; i < al.size(); i++) {***

***System.out.print(“\n”+al.get(i));***

7) ***try {***

***FileReader fr = new FileReader(args[0]);***

***BufferedReader br = new BufferedReader(fr);***

***int ch;***

***while( (ch = br.read() ) != -1) {***

***System.out.println(ch);***

***}***

***}***

***catch(NullPointerException e){***

***System.out.println(“usage: command <file>”);***

***return; // optional***

***}***

***catch(FileNotFoundException e){***

***System.out.println(“File not found”);***

***return; // optional***

***}***

***catch(IOException e){***

***System.out.println(e.getMessage());***

***return; // optional***

***}***

8)

