Tile Designer Project

Where is your project located?

G:\My Drive\ICS4U0-B Gurpreet Singh\Eclipse Workspace\Unit 3 - Inheritance

What is the name of the Java class file that is intended for the customer to use (includes the menu):

Client.java

What is the name of the black box test case Java class where you test out every feature of your application:

Testing.java

Instructions:
For the black box Java file, put all of your test cases into one Main program and put breaks in between each test case after the output is shown.
Example:

```
Scanner input = new Scanner (System.in);
System.out.println("This is the output for test case #1.  Press ENTER to show the next test case.");
input.next();
System.out.println("Starting test case #2:");
//rest of code
```

Test Cases: (You will need to add more rows to show every possible test case to prove that your program works)

Test Case 1:	**Description:**	**Code:**	**Output:**

Tile Designer Project

	This code tests for all the methods and variables in the Rectangle class including width, length, Area (), Perimeter () and toString()	`System.out.println("Case 1 - Testing Triangle class");`
		`Rectangle rectangle = new Rectangle ("red", 3,3);`
		
		`System.out.println(rectangle.toString());`
		
		`System.out.println("Area - " + rectangle.Area());`
		
		`System.out.println("Perimeter - " + rectangle.Perimeter());`
			`Case 1 - Testing Triangle class`
			`RE l: 3.0 w: 3.0 c: red`
			`Area - 9.0`
			`Perimeter - 12.0`

Test Case 2:	This code tests for all the methods and variables in the Square class including width, length, Area (), Perimeter () and toString()	`System.out.println("Case 2 - Testing Square class");`
		`Square square = new Square ("blue", 5);`
		
		`System.out.println(square.toString());`
		
		`System.out.println("Area - " + square.Area());`
		
		`System.out.println("Perimeter - " + square.Perimeter());`
			`Case 2 - Testing Square class`
			`SQ s: 5.0 c: blue`
			`Area - 25.0`
			`Perimeter - 20.0`

Test Case 3:	This code tests for all the methods and variables in the Trapezoid class	`System.out.println("Case 3 - Testing Trapezoid class");`
		`Trapezoid trapezoid = new Trapezoid ("pink", 55, 15, 30, 20);`
			`Case 3 - Testing Trapezoid class`
			`TR a: 55.0 b 15.0 c: 30.0 d: 20.0 h: 12.0 c: pink`
			`Area - 9900.0`
			`Perimeter - 120.0`

Tile Designer Project

	including long side, short side, side1, side2, Area (), Perimeter () and toString()	`System.out.println(trapezoid.toString());`
		
		`System.out.println("Area - " + trapezoid.Area());`
		
		`System.out.println("Perimeter - " + trapezoid.Perimeter());`	

Test Case 4:	This code tests for all the methods and variables in the Ellipse class including horizontal radius, vertical radius, Area(), Circumference() and toString()	`System.out.println("Case 4 - Testing the Ellipse class");`
		`Ellipse ellipse = new Ellipse ("blue",7,6);`
		
		`System.out.println(ellipse.toString());`
		
		`System.out.println("Area - " + ellipse.Area());`
		
		`System.out.println("Circumference - " + ellipse.Circumference());`
			`Case 4 - Testing the Ellipse class`
			`EL a: 6.0 b: 7.0 c: blue`
			`Area - 131.88`
			`Circumference - 40.94059110467264`

Test Case 5:	This code tests for all the methods and variables in the Circle class including radius, Area(), Circumference() and toString()	`System.out.println("Case 5 - Testing the Circle class");`
		`Circle circle = new Circle ("yellow", 5);`
		
		`System.out.println(circle.toString());`
		`System.out.println("Area - " + circle.Area());`
		
		`System.out.println("Circumfer`
			`Case 5 - Testing the Circle class`
			`CI r: 5.0 c: yellow`
			`Area - 78.5`
			`Circumference - 31.400000000000002`

Tile Designer Project

		ence - " + circle.Circumference());	
Test Case 6:	This code tests for all the methods and variables in the Half-Circle class including radius, Area(), Circumference() and toString()	System.out.println("Case 6 - Testing the Half-Circle class"); HalfCircle halfcircle = new HalfCircle ("violet", 3); System.out.println(halfcircle.toString()); System.out.println("Area - " + halfcircle.Area()); System.out.println("Circumference - " + halfcircle.Circumference());	Case 6 - Testing the Half-Circle class HC r: 3.0 c: violet Area - 14.13 Circumference - 15.420000000000002
Test Case 7:	This code tests for all the methods and variables in the Donut class including inner radius, outer radius, Area(), Circumference() and toString()	System.out.println("Case 7 - Testing the Donut class"); Donut donut = new Donut ("orange", 6, 8); System.out.println(donut.toString()); System.out.println("Area - " + donut.Area()); System.out.println("Circumference - " + donut.Circumference());	Case 7 - Testing the Donut class DO i: 6.0 o: 8.0 c: orange Area - 87.92 Circumference - 50.24

Tile Designer Project

Test Case 8:	This code tests for the printTile () method when the ARRAY IS EMPTY. When no tile is added this method is supposed to display the 2D array populated with the word “EMPTY”	`System.out.println("Case 8 - Testing the printTile() method in the Room class when ROOM IS EMPTY");`
		`room.printTile();`
			`Case 8 - Testing the printTile() method in the Room class when ROOM IS EMPTY`
			`EMPTY EMPTY`
			`EMPTY EMPTY`
			`EMPTY EMPTY`

Test Case 9:	This code tests for the insertTile(0 method in the Room class. This method takes in the row and column indexes and inserts the tile in the 2D array at the specified locations.	`System.out.println("Case 9 - Testing for insertTile() Method in Room class");`
		`Tiles tile = new Square ("blue" , 5);`
		`room.insertTile(tile, 1, 1);`
		`System.out.println("Tile Inserted");`
			`Case 9 - Testing for insertTile() Method in Room class`
			`Tile Inserted`

Test Case 10:	This code tests for the printTile () method when a TILE IS INSERTED in the 2D array. When a tile is added this method is	`System.out.println("Case 10 - Testing for printTile() Method when TILE IS INSERTED");`
		`room.printTile();`
			`Case 10 - Testing for printTile() Method when TILE IS INSERTED`
			`SQ s: 5.0 c: blue EMPTY`
			`EMPTY EMPTY`
			`EMPTY EMPTY`

	supposed to display the toString() method of tile at the specified location by the user.		

Test Case 11:	This code tests for the removeTile () method in the Room class This method takes in the row and width indexes as parameters and removes the tile at the specified location. If the location is already empty it lets the user know that the location specified is empty.	`System.out.println("Case 11 - Testing for removeTile() Method in Room class");`
		`room.removeTile(1,1);`
		`System.out.println("Tile Removed");`
			`Case 11 - Testing for removeTile() Method in Room class`
			`Tile Removed`

Test Case 12:	This code tests for the printTile () method when a TILE IS REMOVED in the 2D array. When a tile is	`System.out.println("Case 12 - Testing for printTile() Method in Room class when TILE IS REMOVED");`
		`room.printTile();`
			`Case 12 - Testing for printTile() Method in Room class when TILE IS REMOVED`
			`EMPTY EMPTY`
			`EMPTY EMPTY`
			`EMPTY EMPTY`

Tile Designer Project

	removed this method is supposed to display “EMPTY” at the location where the tile was deleted.		

Test Case 13:	This code tests for the totalArea() method in the Room class.This method is supposed to calculate the sum of the area of all tiles in the 2D Array and display it to the user.	`System.out.println("Case 13 - Testing for totalArea() method in Room class");`
		`Tiles rectangle1 = new Rectangle ("yellow", 4,4);`
		`Tiles circle1 = new Circle("maroon", 5);`
		
		`room.insertTile(rectangle1, 1, 1);`
		
		`room.insertTile(circle1, 2, 2);`
		
		`System.out.println("Total Area - " + room.totalArea() + " feet");`
			`Case 13 - Testing for totalArea() method in Room class`
			`Total Area - 94.5 feet`

Test Case 14:	This code tests for the totalCost() method in the Room class.This method is supposed to	`//Test Case 14: Testing for totalCost() method in Room class`
		
		`System.out.println(" Case 14 - Testing for totalCost() method in Room class");`
			`Case 14 - Testing for totalCost() method in Room class`
			`Total Cost - $236.25`

Tile Designer Project

	calculate the total cost of all the tiles in the 2D Array and display it to the user. Make sure that you set the price for each tile by calling the method setCost() and passing the cost of each tile as a parameter before you call the totalCost() method/	`room.setPrice(2.50);`
		
		`System.out.println("Total Cost - $" + room.totalCost());`	

Test Case 15:	This code tests for the RandomTile() method in the Room class. This method is supposed to generate random tiles and populate the entire 2D Array with random tiles.	`System.out.println("Case 15 - Testing for RandomTile() method in Room class");`
		`HashSet<String> colourList = new HashSet<String>();`
		`colourList = ColourList();`
		`Tiles tile2 = null;`
		`Room room2 = new Room (3,3);`
		`double maxWidth = 3;`
		`double maxLength = 3;`
		`int rowIndex;`
		`int columnIndex;`
		`boolean valid;`
			`Case 15 - Testing for RandomTile()  method in Room class`
			`Random tiles generated successfully`

Tile Designer Project

		while (room2.RoomStatus() == false) { tile2 = room2.RandomTile(maxWidth, maxLength, colourList); do { rowIndex = (int) ((int) 1 + (Math.random() * (3))); columnIndex = (int) ((int) 1 + (Math.random() * (3))); valid = room2.insertTile(tile2, rowIndex, columnIndex); } while (valid == false); } System.out.println("Random tiles generated successfully");	
Test Case 16:	This code tests for the printTile() method in the Room class when the RANDOM TILES ARE GENERATED. When random tiles are	System.out.println("Case 16 - Testing for printTile() Method in Room class when RANDOM TILES ARE GENERATED"); room2.printTile();	Case 16 - Testing for printTile() Method in Room class when RANDOM TILES ARE GENERATED EL a: 1.7 b: 3.8 c: khaki EL a: 1.4 b: 3.9 c: ivory SQ s: 3.1 c: coral RE l: 3.0 w: 3.6 c: pink EL a: 1.5 b: 1.3 c: black CI r: 2.6 c: ivory

Tile Designer Project

	generated this method is supposed to display all the random tiles that were generated by the Random Tile () method.		

Which additional features did you implement to achieve a level 4-/4/4+ ? If none, please leave this blank.

The additional feature that I have added into my program is when the user decides to remove a tile from an empty row/column my user lets the user know that the specified row and column is already empty.

What do you think your project deserves overall in terms of the overall mark?

Categories	**Level 1** **(50 – 59%)**	**Level 2** **(60 – 69%)**	**Level 3** **(70 – 79%)**	**Level 4** **(80 – 100%)**	
Communication Code Design Program Header Comments Variable names Method names Indentation	Variable names and method names along with a description of methods and any required parameters are missing crucial elements and do not communicate a clear plan	Variable names and method names along with a description of methods and any required parameters is communicated with limited clarity	Variable names and method names along with a description of methods and any required parameters is communicated with only minor omissions or errors	Variable names and method names along with a description of methods and any required parameters is communicated clearly	

Tile Designer Project

/ 10					
Communication Throughout my code I made sure that the variables and methods names are effectively explained in a simplified manner so that the person looking at my code can easily understand the algorithm and procedure of my code. In addition, to further my communication on all the classes, I made sure to add the authors name (myself), date and a brief description describing the functionality of the classes so that the person looking at my classes has a clear understanding on what each class performs. Overall, I believe that I deserve a 100% based on initiatives I took to clearly and effectively communicate my assignments.					
Thinking Program Testing Black Box Testing /10	Program testing was missing crucial elements.	Program testing was insufficient to conclude that the program runs properly	Program testing missed covered a considerable number of possible cases and was summarized in a logical way	Program testing was thorough, and summarized in a logical and succinct way	
Thinking As you can tell from my test cases above, while I was programming my assignment I constantly tested and debugged my program in order to avoid my program to behave unexpectedly and/or crash in the middle while the user is running it. Therefore I made sure that I have performed every test case possible so that the user can have an excellent experience with my program. Furthermore, all my test cases were summarized and explained in the most simplest and effective way possible so that a non-programmer can also understand the algorithm and procedures that I took in order to test my program.. In conclusion, I believe that I deserve a 100% on the thinking part of the project as I have tested every single scenario possible throughout my test cases and summarized them in an logical and succinct way					
Application Program Execution Error Checking	Program doesn't run properly, or the output has serious errors.	Program runs properly. Output has errors. The provided client code will not run properly, but student provided client does run properly	Program runs properly. Output is correct with only minor errors. The provided client code will run properly	Program runs properly as is. The output is identical or superior to the sample provided. The provided client code will run properly with an additional feature.	

Tile Designer Project

/ 25					
Application In the process of programming my Tiles Designer project, I have worked day and night into checking any errors that my program might have. I constantly made sure that I give the user an marvelous experience by ensuring that my program runs smoothly and it is laid out in a well structured manner. To add on, in order to make my program even more efficacious, I have added 1 additional feature which works when the user decides to delete a tile that is already empty, however my program triggers a message letting the user know that the specified index is already empty. Furthermore, I have also added ASCII-art in order to make the program look more appealing and alluring. Overall, judging by the execution of my program and the overall smoothness and appeal, I strongly believe that I deserve a 100% in this section.					
Knowledge UML Diagram Programming Concepts <u>Focus:</u> Inheritance Polymorphism Abstract Class / 25	Few of the programming concepts from the unit are used properly	Some of the programming concepts from the unit are used properly	Most of the programming concepts from the unit are used properly	Many or all of the programming concepts from the unit are used properly to maximize the efficiency of the code	
Knowledge My initial goal from the beginning of this assignment was to not only cover up the concepts that we have learned in this unit but to also include concepts from all the other units from Object oriented programing to Inheritance. Now that the project is done, I can proudly look at my project and say that I have hit every major concept that we have learned in this unit including Polymorphism abstract classes, inheritance and much more.In addition, I made sure that all of my programs are coded in an efficient and reliable manner in order to reduce resources consumption and preventing my program from slowing down. In conclusion, I believe that I deserved 97% in the knowledge section of this assignment judging by coverage of most of the programming concepts from this unit as well as the efficiency and reliability of my programs.					

Is there anything else you would like me to know about your project or progress in the course?

Dear Mr Di Iorio, throughout my high school years I was fortunate enough to always have you as either my Computer Science or Computer Engineering teacher. Within all these years, I had many teachers however you were one of those few teachers that helped me discover my true passion in life, which is programming & coding, and if it wasn't

Tile Designer Project

because of your officious teaching style, this hidden passion that I had within me would have never woken up. Particularly, back in Grade 10, I was clueless on what I wanted to pursue in my life and this would constantly stress me out knowing the fact that I did not have any future plans or even a purpose to my life. However, this did not last for long, as the moment I stepped into your Computer science class, I instantly fell in love with the entire concept of Computer Science and that is exactly when I found my hidden passion and interest for programming. What made Computer Science more interesting and captivating for me, was your efficacious teaching style and your ability to explain things effectively but in an simplistic manner in which everyone could understand. Long story short, I simply wanted to thank you from the bottom of my heart for helping me discover something that I crucially love and giving me a clear vision onto my career path. I will always thank you for all the extra effort you made to help me grow as well as all the challenges that you encouraged me to face to not only help me become a better programmer but to also help me become a better person in the real world. You're not just my teacher, you're also my mentor, authority and guide all rolled into one person and I will always be grateful for your presence in my life.