Item	Possible Points	Earned Points	Notes

Program design for Project
 	
• Uses ConsoleTestApp to test the Console class.
 	
• The Console class has the following methods:
```// for output```
```public void print(String s);```
```Public void println(String s);```
```public void println();```
 	
```// for input```
```public String getRequiredString(String```
```   prompt);```
```public String getChoiceString(String prompt,```
```   String s1, String s2);```
```public int getInt(String prompt);```
```public int getIntWithinRange(String prompt,```
```   int min, int max);```
```public double getDouble(String prompt);```
```public double getDoubleWithinRange(String```
```   prompt, double min, double max);```
 	
• The Console class is stored in a package named `myUtils.util`
• The Console class uses javadoc comments including the `@param` and `@return` tags.
	5		

Design Diagrams:
• A correct class diagram is provided for all classes
• Design documentation reflects actual logic of code
• All methods are documented (one diagram for each method; you may have more than one diagram on a page)
• No diagram is larger than one page (8 ½ by 11 inches with ½ inch margins on all sides)
• If using flowcharts to diagram the logic:
  o Each flowchart begins and ends with a terminator symbol Note: the main method beginning terminator contains the word `main()`. The main method ending terminator contains the word `return`. Because you do not write the code that calls the main method, you will not have any flowcharts where the beginning terminator contains the word START and the ending terminator contains the word END.
  o The appropriate symbol is used
  o Only one task per process symbol (the rectangle); each variable declaration should be in its own symbol; show the entire formula for calculations
  o Every symbol (except a terminator) has at least one flowline leading to it and one and only one flowline leading from it.
• If using structured pseudocode to diagram the logic:
  o The pseudocode is appropriately indented
  o Each variable declaration is on its own line
  o The entire formula is shown for calculations
  o Selection and iteration blocks have a clear beginning and ending
• If using Warnier Diagrams to diagram the logic:
  o Braces are appropriately labeled
  o Each variable declaration is on its own line
  o The entire formula is shown for calculations
	5		

Item	Possible Points	Earned Points	Notes

Following course standards:
• Code standards:
  o Code restricted to 80 columns
  o Follows naming conventions for classes, variables, methods, and constants
  o Appropriate comment block at top of program file (may use javadoc conventions)
  o Methods appropriately commented (may use javadoc conventions)
  o Variables have meaningful names
  o Braces align correctly
  o Control statements formatted correctly
• All non-code files contain your name, the course code (CITP 190), and the project number at the top of the file.
• All design diagrams are in one file.
• All files are in standard 8 ½ by 11 inch format with at least ½ inch margins on all sides of the page.
	5		

Proof • Shows all test data	3		
Screen captures	2		
Penalties: • Incorrect calculations • Output is not presented as shown (including spelling and spacing) • Code does not follow the standards • Not all test data was used • Reflects material outside what has been covered through Chapter 9 • Using any classes not mentioned in the instructions or does not use one of the classes mentioned in the instructions • Using a continue statement or misusing a break statement	**-20 for any of the items listed**		
Total	**20**	**0.0**	