**Unit 5 Programming Problems Worksheet**

# Programming Problem 1 – CycleThrowTryCatch

Revisit the Cycle class in Module 3.  Modify your application such that the properties, **numberOfWheels** and **weight** are entered as **double** values interactively (at the keyboard).  Exception handling will be used to determine whether a type mismatch occurs.

Edit your application such that, in addition to [A], the values for **numberOfWheels** and **weight**, entered interactively, will throw a new exception “Values cannot be less than or equal to zero” only If the values are less than or equal to zero.  Add or use the appropriate try and/or catch blocks.

**Directions**

Examine your application for the class called **Cycle**.

* Add Try and Catch blocks appropriately.
* Add the **throw statement for the new exception.**
* Display an appropriate message if an exception occurs.
* Display the properties of the object.

**Grading Rubric**

|  |  |
| --- | --- |
| **Task** | **Points** |
| Try and Catch blocks added correctly | 1 |
| Appropriate message used for each exception | 1 |
| Throw statements used correctly | 1 |
| Object properties displayed correctly | 1 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **6** |

# Screenshots

# Programming Problem 2 – CycleFileOutput

Revisit the Cycle class in Module 3.  Modify your application such that the properties will be written to a text file called “Cycle.txt” instead of to the screen.

**Directions**

Examine your application for the class called **Cycle**.

* Add an appropriate throws statement in the main method.
* Create a reference to a File class with the appropriate name of a text file (Cycle.txt).
* Use appropriate code to ensure that the text file exist.
* Output the values of the variables to the text file.
* Close the file.

**Note**: Verify the contents were written to the text file using notepad (or any word processor).

### Grading Rubric

|  |  |
| --- | --- |
| **Task** | **Points** |
| Throws clause added in main method | 1 |
| Create a reference to the File class and text file | 1 |
| Check whether the text file exists | 1 |
| Output the properties to the text file | 1 |
| Close the text file | 1 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **7** |

# Screenshots

# Programming Problem 3 – CycleFileInput

Revisit the Cycle class in Module 3.  Modify your application such that the properties will be read from a text file called “Cycle.txt”.

**Directions**

Examine your application for the class called **Cycle**.

* Add an appropriate throws statement in the main method.
* Create a reference to a File class with the appropriate name of a text file (Cycle.txt). Note: Cycle.txt was created in the previous assignment, CycleFileOutput.
* In your code, check that the text file does exist.
* Input the values from the file to memory.
* Close the file.

### Grading Rubric

|  |  |
| --- | --- |
| **Task** | **Points** |
| Throws clause added in main method | 1 |
| Create a reference to the File class and text file | 1 |
| Check whether the text file exists | 1 |
| Read the properties from the text file and output to screen | 1 |
| Close the text file | 1 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **7** |

# Screenshots