**Unit 8 Programming Problems Worksheet**

# Programming Problem 1 – Fraction

Consider the following class:

public  class Fraction{
                private int numerator;
                private int denominator;
                private static char slash = ‘/’,
}

Write a program to instantiate an object of the Fraction class to test the class..  Add a constructor(s), set and get methods to the Fraction class appropriately.  Using the Serializable Interface, instantiate and write three Fraction objects to a file called “SerialF.dat.”  A loop must be used when creating and writing the three objects to the file. You can select the values you wish to use for the denominator and numerator.

**Directions**

* Create the Fraction Class.
	+ Add a constructor(s), set and get methods to the Fraction class appropriately.
	+ Ensure the class is Serializable.
* Use a loop to create three Fraction objects.
	+ Initialize the three objects,
	+ Write each object to the file“SerialF.dat.”
	+ Display an appropriate message if an exception occurs.

**Grading Rubric**

|  |  |
| --- | --- |
| **Task** | **Points** |
| Fraction class implements Serializable | 0.5 |
| The main method throws an IOException | 1 |
| The file "SerialF.dat" is created properly | 1 |
| Loop used to instantiate objects | 1 |
| Objects are correctly written to file "SerialF.dat" | 1 |
| The file "SerialF.dat" is closed appropriately | 0.5 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **7** |

# Screenshots

# Programming Problem 2 – ReadMe

The pledge of Allegiance states, “I pledge allegiance to the Flag of the United States of America, and to the Republic for which it stands: one Nation under God, indivisible, With Liberty and Justice for all.” Save the pledge, 174 characters, to a text file.  Use the RandomAccessFile class to access the file.  Using Seek method, display the characters at positions 124 and 135 only.

**Directions**

* Create a text file with the pledge of allegiance named Pledge.txt using notepad.
* Create a reference to the file, Pledge.txt, using the RandomAccessFile class.
* Use the Seek method to point at positions 124 and 135, then display the characters represented by the byte streams.
* Use try, catch and finally blocks for exception handling.

### Grading Rubric

|  |  |
| --- | --- |
| **Task** | **Points** |
| Throws clause added in main method of the ReadMe class | 1 |
| Create a reference of the RandomAccessFile class which points to Pledge.txt | 1 |
| Include try, catch and finally blocks for exception handling | 1 |
| Use the Seek method to point at positions 124 and 135, then display characters represented by the byte streams | 1 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **6** |

# Screenshots

# Programming Problem 3 – ReadWrite

Write an application which will write five student ID numbers and GPAs to a “rw” file called “Stu.dat” and then allow you to display the GPA of any student upon entering their ID number, for any number of students.

**Directions**

* Import the classes necessary to support your application.
* Create a class called ReadWrite.  This class has no properties or behaviors.
* Create a main method which will include the following:
	+ Add an appropriate throws statement in the main method.
	+ Create a reference to a text file called “Stu.dat” with “rw” access.
	+ Include try and catch blocks for exception handling.
	+ Use a loop to interactively assign student ID numbers and their GPA scores.
	+ With the use of a second loop, display the GPA for specific student ID numbers.
	+ Use a sentinel to determine when you wish to stop the program.
	+ Include a finally block within your program.

### Note: User defined methods are not required but may be used if you prefer.

### Grading Rubric

|  |  |
| --- | --- |
| **Task** | **Points** |
| Throws clause added in main method | 0.25 |
| Create a reference to the RandomAccessFile class | 1 |
| Include try and catch blocks for exception handling | 1 |
| Use a loop to interactively assign student ID numbers and their GPA scores | 1 |
| Use of a second loop to display the GPA for specific student ID numbers | 1 |
| Use of a sentinel to determine when you wish to stop the program | 0.25 |
| Include a finally block within your program | 0.5 |
| Proper documentation | 1 |
| Program works effectively | 1 |
| **Total** | **7** |

# Screenshots