COMP 1030

Lab #8

Multi-dimensional arrays & Exceptions

Introduction

During this assignment you will build two java classes. The first class will contain the required state and behaviour for the object, **but NO main method**. The second class will contain simply the main method to give the JRE an entry point into the program, a line to instantiate a new object based upon the first class and a few lines to exercise the functionality of the first class.

When writing your code, keep these guidelines in mind:

- Start each class with the proper javadoc comment header. The first line of that comment should be the purpose of the class.
- Provide a comment for **each** section of code as well as a javadoc header.
- Follow the layout for your class as illustrated below:

Javadoc comment header
Import statements (if required)
Class declaration
    State (instance variables/data)
    Constructor(s) (if required)
    Behaviour(s) (method(s))
Close class declaration

- Use whitespace and indentations to make your code more readable and easy to debug.
- Be sure to clearly understand your work – do not simply copy code from someone else.

Instructions: Use any IDE to complete this assignment other than a simple notepad or BlueJ.

Create a class that acts as a blueprint for a College Course.

- This class should have the following state:
    - int – course reference number
    - int - course number
- This class should have the following behaviour:
    - Appropriate setters and getters
- This class should have the following constructors:
    - Default constructor
    - Constructor that takes two arguments.

Create a second class called CollegeCourseTestHarness:

- Create a two dimensional array with 10 columns and 200 rows.
- Fill the first 100 rows with CollegeCourse Objects in which the course reference number and course number are set to the default value of zero.
- Fill the second 100 rows with CollegeCourse Objects in which the course reference number and course number are set to the default value of 9999.
- Iterate the array and print out the state of each object with an appropriate short label. Print a dashed line between each data set.
- Iterate the array and set the state of each object as follows:
    - Set the course reference number to a sequential/unique 7 digit value
    - Set the course number to one of four, 5 digit course numbers that are randomly chosen from a list you make up.
- Iterate the array and print out the state of each object with an appropriate short label. Print a dashed line between each data set.
- Ask the user for a column and row number. Print out the state of the object corresponding to this location. Take into account the probability that the user will provide a value that is out of the bounds of the array, which will throw an exception. Handle this exception in your code using a try/catch block to ensure your program will not crash but instead will inform the user that they need to provide a valid set of numbers.

Things to consider for success:

- Follow the layout for your class as illustrated above.
- Write POC code before tackling the full problem.
- Write a section at a time and compile after each section so you do not have a volume of compiler errors to deal with.
- Comment as you go.
- Use indents and whitespace appropriately to make your code readable.
- Be sure that you actually understand the code you have written, simply copying someone else's code will not aid in your understanding of the java language.
- Stay focused and work diligently, collaborate with others if you are stuck.