CS-0401 Spring 2020 Final Exam

General Exam Info:

a) All your answers shall be provided by code samples inside one single Netbeans project, the project name shall be CS_0401_Final_Exam_<your Pitt ID here>. Please place your full name in a comment line too.
b) Create a Java file for each question below, under the name Question_XY, where XY is the question number, such as 01. Each Question_XY class shall have a main() method, so I can run it to check your answer for the given question.
c) Your code must be compilation error-free.
d) If you need to create additional classes for one given question, name them "Question_01_AnyNameHere". Example Question_01_Banana
e) Don't overshoot on your answers. Just implement the minimal code that answers a given question.
f) When finished, export this project (zip file in Netbeans) and send it to me asap (brasko@pitt.edu).
g) **Exam deadline is Wednesday, April 22nd, 8:00 pm**. No exams will be accepted after the due date.
h) Answers will be checked with internet sites and classmates. Copying solutions from the internet or classmates will grant you an F (and the classmate too, if the case).

To start your exam, go to Netbeans and create the following Java application project:

CS_0401_Final_Exam_<your Pitt Id here>

Example:

CS_0401_Final_Exam_abc123

Question 1:

a) Create a sample code that demonstrates the use of inheritance
a) Put in comment lines the advantages of using inheritance in a program
b) Create an example of chain of inheritances. Use the code used above as a starting point for this question. The top most super class of this chain of inheritances must have only one constructor. This constructor must be a non-default.
c) In Question_01 java file, create instances of the classes that you have developed.

Question 2:

a) Create an example that demonstrates the difference between abstract and interfaces. Again, don't make it too complex, just create java classes that use both concepts.
b) Place in comment lines the differences from abstract classes and interfaces.

Question 3:

a) Create an example that uses Wrapper classes to convert the following arrays into Java Lists

```
int[ ] sizes;
double[ ] prices;
Car[ ] arrayOfCars;
```

b) Initialize the lists for sizes and prices with some numbers
c) Create methods that compute the min, max, average values for the prices list using regular FOR loop and Enhanced FOR loops

Question 4:

a) Based on the code below, create an example that demonstrates the use of Exceptions. One showing handling the exception inside a method and other throwing the exception up to the caller method.

```
public void manipulateArray( double[ ] array, int number) {
        for (int i=0; i<number-1; i++) {
                System.out.println("array element: " + array[i]);
                System.out.println("some calculation results: " + 1.0/(array[i] - array[i+1]);
        }
}
```

b) Call this method three times:
i) With array set to null and number set to 10;
ii) with array set to { 1, 3, 5, 7, 9} and number set to 10;
iii) with array set to { 1, 2, 3, 4, 5} and number set to 4;

c) If there are more than one exception to handle, use the try-catch for one and the "throws" for the other
d) Explain with code or in comment lines how polymorphism can be used with Exceptions.

Question 5:

Based on the equation below,

$$\frac{1}{n} + \frac{2}{(n-1)} + \ldots + \frac{n}{1}$$

a) Create methods that compute the value of this equation based on an input argument n by:
- Using a FOR loop
- Using recursion

b) Call those methods from the main class and print the results in the terminal window.

Question 6:

Create a program that exemplifies the concept of method overload and method override. If needed add comment lines to further explain the idea behind your code (only if you think you need it. It is not required).

Question 7:

a) Create a program that exemplifies the difference of calling static and non-static methods.

Question 8:

a) Create an enumeration (any name and some items inside)
b) Create an object of that enumeration
c) Create a method that accepts an enumeration item as input argument and with an IF-ELSE-IF structure, prints something about that input argument
d) Same thing as item (c) above, but using SWITCH structure

Final thoughts:

- Congratulations! You should be proud of yourself for going through so many OOP programming concepts! We went through about 700 pages of your textbook!
- Thank you for a great semester! Thanks for your participation in class and questioning attitude and for all the fun comments we had!
- Keep studying hard for a little bit more! College will be over in few years for you, and then you will be on your way for a great future, fortune, and wealth! :)
- Keep in touch: the best e-mail to reach me as a former student is paulobrasko@gmail.com. Keep me informed of your success.
- If at some point in time you feel stressed, try to stay calm and remember that every one of us has problems (including me, multiple times a year!). Facing big issues in our lives is part of being human. Instead of feeling sorry about yourself for a long period of time, face your problem as a personal challenge! If you still are stuck after a while, ask for help from friends, parents, teachers, coworkers, etc.
- And finally, and the most important thing: Enjoy life! Work hard, but remember to pursue happiness and be kind to others.

If you want to say something back to me, create a final Java class called MyComments.java.

Thanks