### Hands-On MidTerm TEST

Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Be sure to read the following general instructions carefully:**

* This lab test must be completed individually by all the students.
* Save your program periodically just in case that your PC crashes.

##### Exercise 1

Start a new Eclipse Java project. Name the project: **YourFullName\_COMP228\_003Test**. For example: JohnSmith\_COMP228\_003Test. **DO NOT** create a package in this project. Let Eclipse create the default package.

Create an abstract class called *Book*. The class should declare the following variables:

* an instance variable that describes the *title* - String
* an instance variable that describes the *ISBN* - String
* an instance variable that describes the *publisher* - String
* an instance variable that describes the *price* - double
* an instance variable that describes the *year – integer*

Provide a toString() method that returns the information stored in the above variables.

Create the **getter** and **setter** methods for each instance variable except *price*. Provide the necessary constructors. Include *an* ***abstract*** *method* ***setPrice(double price)*** *to determine the price* for a book. Include an abstract method **getGenre()** to return the genre of the book.

Create two subclasses called *ScienceBook* and *ChildrenBook*.

Book

ChildrenBook

ScienceBook

These subclasses should override the abstract methods *setPrice* and *getGenre* of class *Book*.

Use the following rule for setting the price for a book:

* + science books will have a 10% discount per each book
  + children books will have a fixed price (specified by user).

Write a driver program (another class with **main** method) that uses the above hierarchy. In your driver program you must implement an interaction with the user.

* Use showInputDialog method to let the user input book information.
* Use showMessageDialog method to display book information including price and type for both science and children books.

(10 marks)

You should upload the entire project on eCentennial immediately after demonstration.

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality** |  |
| Correct implementation of classes (instance variable declarations, constructors, getter and setter methods, etc.)  Correct implementation of Inheritance/Polymorphism | 30%  20% |
| Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results) | 35% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Friendly input/output** | 10% |
| **Total** | 100% |