### 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.

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

##### Exercise 1

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

* an instance variable that describes the *title* - String
* an instance variable that describes the *studio* - String
* an instance variable that describes the *leadActor* - String
* an instance variable that describes the *budget* - 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 *budget*. Provide the necessary constructors. Include *an* ***abstract*** *method* ***setBudget(double budget)*** *to determine the budget* for a movie. Include an abstract method **getGenre()** to return the genre of the movie.

Create two subclasses called *ActionMovie* and *ComedyMovie*.

Movie

ComedyMovie

ActionMovie

These subclasses should override the abstract methods *setBudget* and *getGenre* of class *Movie*.

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

* + ActionMovies will have 10% added to their budget
  + ComedyMovies will have a fixed budget (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 movie information.
* Use showMessageDialog method to display book information including budget and type for both action and comedy

(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% |