

COMP1406 - Assignment #1

(Due: Tuesday, January 31st @ 9:00pm)

In this assignment, you will create 5 objects and get them to interact together. You will create theatres for which patrons will buy tickets from a box office to watch movies.



(1) The **Movie**, **Ticket**, **Theatre** and **Patron** Classes

You will need to define 4 objects as indicated below. You must choose appropriate attribute names so that the test program that follows compiles and runs properly.

- Define a class called **Movie** that maintains the title of a movie as well as the amount of earnings it has made since it opened at the theatre.
- Define a class called **Theatre** that keeps track of the **Movie** object that is currently playing in that theatre. A theatre should also have a seat capacity and keep track of how many seats have been sold for the movie playing.
- Define a class called **Ticket** that represents a ticket to go and watch a movie. Each ticket is only valid for a specific **Theatre** object.
- Define a class called **Patron** that keeps track of the age of a person as well as the **Ticket** object that he/she has purchased.

Write any necessary code so that the following test program works as indicated in the output that follows:

```
class TestProgram {
    public static void main(String args[]) {
        Movie m = new Movie("Megamind");
        System.out.println(m.title);
        System.out.println(m.earnings);

        Theatre theatre = new Theatre(3);
        System.out.println(theatre.capacity);
        System.out.println(theatre.seatsSold);
        theatre.moviePlaying = m;

        Patron mary = new Patron(15);
        System.out.println(mary.age);
        System.out.println(mary.ticket);
        mary.ticket = new Ticket(theatre);
        System.out.println(mary.ticket.theatre.moviePlaying.title);
    }
}
```

The expected output is shown here on the right ...
Make sure that your code works before you continue.

```
Megamind
0.0
3
0
15
null
Megamind
```

(2) The **BoxOffice** Class

Define a class called **BoxOffice** that keeps track of two theatres for which it sells movie tickets. The cost of a movie ticket is \$6.25 for children 12 and under, \$5.75 for adults 65 years old or more and \$12.50 for everyone else. The box office should also keep track of the movie that has made the most money in the past.

Carefully examine the test program below. Understand how it is supposed to work. Then write the necessary methods so that the code runs properly, producing the correct results. You may NOT alter the test program ... the TA will be using it to test your code. Note that when tickets are sold to patrons, the title of the movie is provided. You will need to determine which theatre that movie is playing in. As a hint, you can compare two strings using the `.equals()` method in JAVA as follows:

```
String s1 = ...;
String s2 = ...;
if (s1.equals(s2))
    ...;
else
    ...;
```

Note as well in the program below that the **bestMovie()** will either be a previous movie that had the most earnings or one of the current playing movies ... whichever has the most earnings.

Here is the test program:

```
class BoxOfficeTestProgram {
    public static void main(String args[]) {
        // Create a box office with two theatres, each with a capacity of 5 seats
        BoxOffice box = new BoxOffice(5, 5);

        // Open up a couple of new movies at the box office
        System.out.println("Theatre A opens movie: Iron Man");
        box.openMovie("Iron Man", box.theatreA);
        System.out.println("Theatre B opens movie: Zookeeper");
        box.openMovie("Zookeeper", box.theatreB);

        // Now create some patrons
        Patron p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12;
        p1 = new Patron(15);
        p2 = new Patron(26);
        p3 = new Patron(7);
        p4 = new Patron(72);
        p5 = new Patron(65);
        p6 = new Patron(11);
        p7 = new Patron(19);
        p8 = new Patron(17);
        p9 = new Patron(12);
        p10 = new Patron(14);
        p11 = new Patron(13);
        p12 = new Patron(16);

        // ... continued on the next page ... //
    }
}
```

```

// Sell some tickets to the patrons
System.out.println("Patron 1 buys ticket to see Iron Man");
box.sellTicket(p1, "Iron Man");
System.out.println("Patrons 2,3 and 4 buy tickets to see Zookeeper");
box.sellTicket(p2, "Zookeeper");
box.sellTicket(p3, "Zookeeper");
box.sellTicket(p4, "Zookeeper");
System.out.println("Zookeeper seats remaining: " +
    (box.theatreB.capacity - box.theatreB.seatsSold));

System.out.println("\nPatron 5 buys ticket to see The Giggling Giraffe");
box.sellTicket(p5, "The Giggling Giraffe");

System.out.println("\nPatrons 6 and 9 buy tickets to see Zookeeper");
box.sellTicket(p6, "Zookeeper");
box.sellTicket(p9, "Zookeeper");
System.out.println("Patrons 7, 8 and 10 buy tickets to see Iron Man");
box.sellTicket(p7, "Iron Man");
box.sellTicket(p8, "Iron Man");
box.sellTicket(p10, "Iron Man");

System.out.println("Patron 11 tries to buy tickets to see Zookeeper");
box.sellTicket(p11, "Zookeeper");
System.out.println("Zookeeper sold out ? " + box.theatreB.isFull());
System.out.println("\nPatrons 2, 6 and 10 return tickets");
box.returnTicket(p2);
box.returnTicket(p6);
box.returnTicket(p10);
System.out.println("\nPatron 10 tries to return a ticket again");
box.returnTicket(p10);
System.out.println("Zookeeper sold out ? " + box.theatreB.isFull());
System.out.println("Zookeeper seats remaining: " +
    (box.theatreB.capacity - box.theatreB.seatsSold));

System.out.println("\nPatrons 11 and 12 buy tickets to see Zookeeper");
box.sellTicket(p11, "Zookeeper");
box.sellTicket(p12, "Zookeeper");

System.out.println("\nBest movie title: " + box.bestMovie().title);
System.out.println("Best movie earnings: " + box.bestMovie().earnings);

System.out.println("\nTheatre A opens movie: Iron Man 2");
box.openMovie("Iron Man 2", box.theatreA);
System.out.println("Patrons 1,2,7,8 & 12 buy tickets to see Iron Man 2");
box.sellTicket(p1, "Iron Man 2");
box.sellTicket(p2, "Iron Man 2");
box.sellTicket(p7, "Iron Man 2");
box.sellTicket(p8, "Iron Man 2");
box.sellTicket(p12, "Iron Man 2");
System.out.println("\nBest movie title: " + box.bestMovie().title);
System.out.println("Best movie earnings: " + box.bestMovie().earnings);
}
}

```

Here is the expected output (make sure that your code works before you continue):

```
Theatre A opens movie:  Iron Man
Theatre B opens movie:  Zookeeper
Patron 1 buys ticket to see Iron Man
Patrons 2,3 and 4 buy tickets to see Zookeeper
Zookeeper seats remaining: 2

Patron 5 buys ticket to see The Giggling Giraffe
Movie is not currently playing

Patrons 6 and 9 buy tickets to see Zookeeper
Patrons 7, 8 and 10 buy tickets to see Iron Man
Patron 11 tries to buy tickets to see Zookeeper
Movie is sold out
Zookeeper sold out ? true

Patrons 2, 6 and 10 return tickets

Patron 10 tries to return a ticket again
Patron does not have a ticket
Zookeeper sold out ? false
Zookeeper seats remaining: 2

Patrons 11 and 12 buy tickets to see Zookeeper

Best movie title:      Zookeeper
Best movie earnings:  49.5

Theatre A opens movie:  Iron Man 2
Patrons 1,2,7,8 & 12 buy tickets to see Iron Man 2

Best movie title:      Iron Man 2
Best movie earnings:  62.5
```

NOTE: Submit (on WebCT) all **.java** and **.class** files needed to run. You **MUST NOT use packages** in your code, **nor projects**. Submit ALL of your files in one folder such that they can be opened and compiled individually in JCreator. Some IDEs may create packages and/or projects automatically. You **MUST** export the **.java** files and remove the package code at the top if it is there. Do NOT submit JCreator projects either. **JUST SUBMIT the JAVA and CLASS FILES**. Note that if your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment **WELL BEFORE** it is due !

Please NOTE that you WILL lose marks on this assignment if any of your files are missing. You will also lose marks if your code is not written neatly with proper indentation. See examples in the notes for proper style.
