

[**May use one 8.5 × 11 inch sheet of paper for notes.**] Show all of your work clearly in the space provided or on the additional page at the end of the exam. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (6 points) For each of the following expressions, indicate whether or not the `isTrue()` method will be called. If no, explain why not.

1. `true || isTrue()`
2. `true && isTrue()`
3. `false || isTrue()`
4. `false && isTrue()`

2. (10 points) Suppose:

```
int i = 2;  
int j = 17;  
boolean b = true;
```

Indicate 1) the type (`boolean`, `int`, etc) and 2) the value of `x` in the following expressions:

1. `x = (Math.PI + i);`

2. `x = (++i);`

3. `x = (j++ - 2);`

4. `x = (b != !b);`

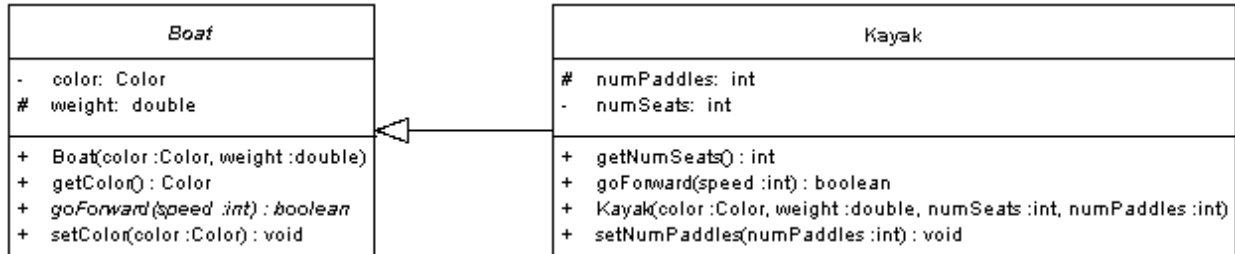
5. `x = (3.0f + 7);`

3. (5 points) Write the first line of a class declaration for a class called `Exam1` that implements an interface named `Refreshable`.

4. (6 points) Implement a class called `Exam1b` that inherits from the class `Exam1` in question **3**. You may assume that `Exam1b` has no additional attributes or methods.

5. (10 points) In as much depth as you can, describe what is returned by the `Object` class's `toString` method?

6. Using this UML class diagram, answer the following questions. **Be sure to justify your answers.**



(a) (10 points) Which of the following expressions is legal (say why):

```

Boat boat1 = new Boat();
Boat boat2 = new Boat(Color.black, 300.2);
Boat boat3 = new Kayak(Color.yellow, 30.2, 1, 1);
Kayak kayak1 = new Kayak(Color.yellow, 30.2, 1, 1);
Kayak kayak2 = new Kayak();
  
```

(b) (2 points) Does an instance of the **Kayak** class have a **color** associated with it?

(c) (2 points) Does an instance of the **Boat** class have a **numSeats** associated with it?

(d) (4 points) List all of the attributes that can be directly accessed within the implementation of the **setNumPaddles** method.

(e) (10 points) Complete the implementation of the **Kayak** classes constructor.

```

public Kayak(Color color, double weight, int numSeats, int numPaddles) {
  
```

7. (15 points) Consider the following interface:

```
public interface Refreshable {  
    /**  
     * Determine if the object is current.  
     * @return Returns true if and only if the object is current.  
     */  
    boolean isCurrent();  
  
    /**  
     * Update or extend the validity period for this object.  
     */  
    void refresh();  
}
```

Suppose that a `static` method, `getRefreshableObject()` from the `ExamI` class exists and that it returns a reference to some kind of object that implements the `Refreshable` interface.

Write a code fragment that will get an object, check to see if the object is current, and refresh the object if it is not current.

8. (20 points) Draw the UML sequence diagram for the following program:

```
public class Exam1 {  
    public static void main(String args[]) {  
        String text = "I love exams!" ;  
        displayMessage(text);  
        displayDate();  
    }  
  
    public static void displayMessage(String message) {  
        String subtext = message.toLowerCase();  
        JOptionPane.showMessageDialog(null, message + subtext );  
    }  
  
    public static void displayDate() {  
        Date today = new Date();  
        SimpleDateFormat formatter = new SimpleDateFormat("EEEE_HH:mm:ss");  
        String dateString = formatter.format(today);  
        JOptionPane.showMessageDialog(null, dateString);  
    }  
}
```



Additional space — identify which problem your work is associated with.