

Name:

Last 4 digits of SID:

Fall 2010 CS151 Object-Oriented Design Midterm II

Instructor: Dr. Kim

Maximum obtainable score: 45 points

38.5

- (5pts) Q1. Define a Java enum type called Seasons to represent SPRING, SUMMER, FALL, and WINTER. A season has an average temperature as an instance variable. (Name the instance variable as temperature.) Write an appropriate constructor and initialize the instance variable with a value of your choice. Also, define methods in a way that the user of the Seasons type should be able to get and set this instance variable.

```

5 public enum Seasons
{
    SPRING(80), SUMMER(90), FALL(70), WINTER(50) }
    private int temperature;
    private Seasons (aTemperature)
    {
        temperature = aTemperature;
    }
    public void setTemperature (int aTemperature)
    {
        temperature = aTemperature;
    }
    public int getTemperature ()
    {
        return temperature;
    }
}

```

- (4pts) Q2. The following program has an error. Explain how you would fix the error.

```

4 public class SuperSubCatch
{
    public static void main(String [] args)
    {
        try
        {
            int a = 0;
            int b = 42/a;
        }
        catch (Exception e)
        {
            System.out.println("Exception.");
        }
        catch (ArithmeticException e)
        {
            System.out.println("ArithmeticException.");
        }
    }
}

```

The proper exception would never get called because the general Exception will always be caught first since it is a superclass to ArithmeticException. It can be fixed by switching around the 2 catch blocks.

Your answer goes here.



(8pts) Q3. Consider the given equals methods of Person and Student, respectively, and answer the questions.

```
class Person
{
    private String name;
    public Person(String n)
    {
        name = n;
    }
    public boolean equals(Object otherObject)
    {
        if (otherObject instanceof Person)
        {
            Person other = (Person) otherObject;
            return name.equals(other.name);
        }
        return false;
    }
}

class Student extends Person
{
    private int id;
    public Student (String n, int i)
    {
        super(n);
        id = i;
    }
    public boolean equals (Object otherObject)
    {
        if (!super.equals(otherObject)) return false;
        if (otherObject instanceof Student)
        {
            Student other = (Student) otherObject;
            return id == other.id;
        }
        return false;
    }
}
```

- (a) What is the output of the following code segment when it is executed with the given equals methods? Write your answer right next to the corresponding System.out.println statement.

```
public static void main (String [] args)
{
    Person p = new Person("Smith");
    Student s = new Student("Smith", 1234);
    System.out.println(p.equals(null)); false
    System.out.println(p.equals(s)); true
    System.out.println(s.equals(p)); false
}
```

- (b) Rewrite the equals method of Student in a way that it satisfies all requirements to be a true equals method. (The equals method of Person should be fixed in a similar manner, but you don't have to write it for this question.)

```
public boolean equals (Object otherObject)
{
    if (this == otherObject) return true;
    if (otherObject == null) return false;
    if (getClass() != otherObject.getClass()) return false;
    if (name.equals(other.name) && id == other.id) return true;
    // use super.equals(other)
}
```

casting?

(7pts) Q4. AbstractSet implements the Iterable Interface. TreeSet is a concrete class that extends AbstractSet. Consider the hashCode method of AbstractSet and the given API and answer the questions.

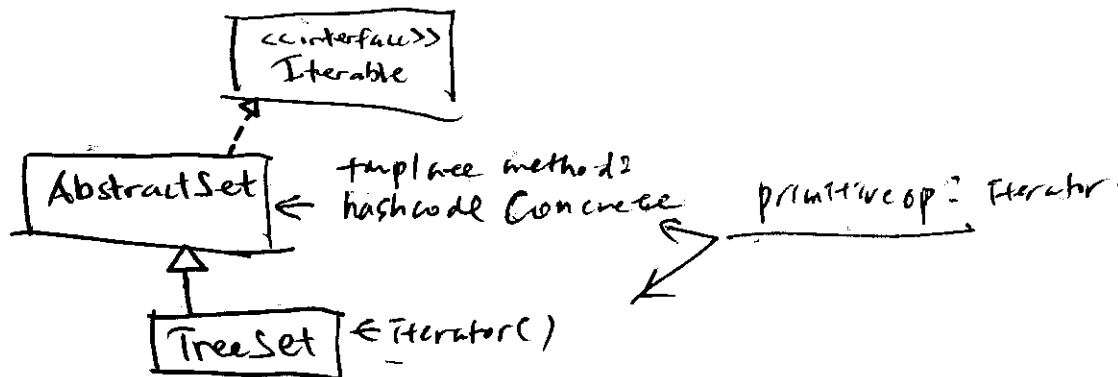
```
public class AbstractSet ...
{
    public int hashCode()
    {
        int h = 0 ;
        Iterator i = iterator();
        while (i.hasNext(h))
        {
            Object obj = i.next();
            if (obj != null) h += obj.hashCode();
        }
        return h;
    }
}
```

Class/Interface	
Iterable<T>	<u>Iterator<T></u> <u>iterator()</u> Returns an iterator over a set of elements of type T.
Iterator<E>	boolean <u>hasNext()</u> Returns true if the iteration has more elements. <u>E</u> <u>next()</u> Returns the next element in the iteration. void <u>remove()</u> Removes from the underlying collection the last element returned by the iterator (optional operation).

(a) Circle the design pattern being used to write the hash code of AbstractSet class.

Strategy Composite Decorator Observer Template Method

(b) Draw a class diagram to depict the design pattern you selected in the part (a). **Do not just copy the class diagram from the book. The class diagram should be specific to this question. (Use names of classes and methods specific to this question.)**



- (7pts) Q5. Consider a window in a windowing system. Suppose the MyWindow class represents a Window without any functionality for adding scrollbars and you want to allow scrolling of the window's content. Note that this question is not about a GUI programming.

```
// the Window interface
interface Window
{
    /** returns a string representation of this window */
    public String toString();
}
```

```
// Implementation of a Window without any scrollbars
class MyWindow implements Window
{
    public String toString()
    { return "Draw this window"; }
}
```

- 1) Which design pattern is suitable to solve the problem ?

Decorator Pattern

- 2) Write one class required to enhance a window object with a scrollbar so that the string representation of the enhanced window with a scrollbar becomes "Draw this window *with a scrollbar*". Follow the design pattern you chose in the part 1). Credit will be given only if you chose a correct answer for the part 1).

```
public class EnhancedWindow implements Window
{
    private Window w;
    public EnhancedWindow(Window aWindow)
    { w = aWindow; }
    public String toString()
    { return w.toString() + "with a scrollbar"; }
}
```

- 3) Write a test program that creates an instance of the enhanced window object and prints its string representation.

```
public class Tester
{
    public static void main (String [] args)
    {
        MyWindow window = new MyWindow();
        EnhancedWindow newWindow = new EnhancedWindow(window);
        System.out.println (newWindow.toString());
    }
}
```

- (6pts) Q6. Consider the classes Employee, Manager, Staff and Test in the packages e, m, s, and t, respectively. When the class Test is compiled, compilation errors are generated. After fixing the errors, the program still throws a run time exception(s). Assume the class Test was compiled and executed with an appropriate classpath option. 1) Circle the erroneous lines and mark them as error1, error2, and error3, respectively. (You need to find three errors.) Note that not all erroneous lines are in the class Test. In the explanation part below, 2) indicate if it is a compilation error or runtime exception and 3) explain the reason of the error. No credit will be given if the explanation is not correct. There will be a penalty if you mark a correct one as an error.

```
package e;
public class Employee
{ private int id;
  protected int getID() { return id; }
}
```

```
package m;
import e.Employee;
public class Manager extends Employee
{ private double bonus; }
```

```
package s;
import e.Employee;
import m.Manager;
public class Staff extends Employee
{
  public boolean IDcompare(Manager m)
  { return getID() == m.getID(); } error 1
}
```

```
package t;
import e.Employee;
import m.Manager;
import s.Staff;

public class Test
{ public static void main (String [] args)
  { Employee[] company = new Manager[2];
    int[] id_set = new int[2];
    for (int i = 0; i < company.length; i++)
      company[i] = new Staff(); error 2
    double[] id_set2 = id_set; error 3
  }
}
```

Your explanations go here.

Error 1: Compilation or Exception (circle one)

Reason of the error: m.getID is protected and cannot be called by Staff because Staff is not a subclass of Manager

Error 2: Compilation or Exception (circle one)

Reason of the error: company objects are of type Manager. Staff is not a manager, so you cannot set Staff to a manager object.

Error 3: Compilation or Exception (circle one)

Reason of the error:

id-set2 is double []

id-set is int []

cannot be set equal to each other

(8pts) Q7. The following program ShapeSelection draws one circle on the screen. When the user selects the circle by pressing the mouse on the circle, the selection of the circle is toggled. The program fills the circle when it is selected. Only outline is drawn for the unselected shape. If the user presses outside of the circle, nothing is changed. **Write the ShapeComponent class** to complete this application. Assume all required library classes are already imported. You don't have to write any import statement.

API and more

1) Create the circle as follows. Note that Ellipse2D.Double is a Shape.

```
new Ellipse2D.Double(10,10, 50,50);
```

2) To add a mouse listener to this ShapeComponent

```
this.addMouseListener(  
    new MouseAdapter()  
    {  
        public void mousePressed(MouseEvent event)  
        {  
        }  
    }  
);
```

Note that addMouseListener is supposed to be inherited from JComponent.

3) Shape: boolean contains(Point2D p)

Tests if a specified Point2D is inside the boundary of the Shape. (Point extends Point2D.)

4) Graphics2D

- public abstract void **draw**(Shape s) strokes the outline of a Shape using the settings of the current Graphics2D context.
- public abstract void **fill**(Shape s) fills the interior of a Shape using the settings of the Graphics2D context.

5) MouseEvent

public Point **getPoint**() returns the x,y position of the event relative to the source component.

6) Header of paintComponent: public void paintComponent(Graphics g)

```
public class ShapeSelection  
{  
    public static void main(String[] args)  
    {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        ShapeComponent scene = new ShapeComponent();  
  
        frame.add(scene, BorderLayout.CENTER);  
        frame.setSize(200, 100);  
        frame.setVisible(true);  
    }  
}  
// Your answer goes in the next page.
```