You can't Disapparate on Hogwarts grounds! - Hermione
Do not Disapparate. Stay calm, and finish the test.
Good luck!

<table>
<thead>
<tr>
<th>Question</th>
<th>Part</th>
<th>Total</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(a)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(d)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(e)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(a)</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b)</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Name: _____________________________________________
[1] The following questions refer to the webpage shown in Appendix A.

Answer the following True / False questions (circle the correct answers): (10 points)

(i) The page can be implemented using a table with three columns [ T / F ]
(ii) More than one house can be selected at any given time [ T / F ]
(iii) The source code did not define a <TITLE></TITLE> [ T / F ]
(iv) When you click on Reset, any selected button(s) gets de-selected [ T / F ]
(v) A <ul> command was definitely used in this page [ T / F ]

[2] The following questions refer to the code documented in Appendix B. The code calculates the radius of a circle in feet and inches, given the circle’s area (in square feet).

(a) Answer the following True / False questions (circle the correct answers): (10 points)

(vi) RoundObject is a superclass of Circle [ T / F ]
(vii) The one-parameter Circle constructor calls the one-parameter RoundObject constructor [ T / F ]
(viii) The method displayFeetAndInches requires a target instance [ T / F ]
(ix) The three different Circle constructors are examples of encapsulation [ T / F ]
(x) The Circle class must have a method that calculates the radius [ T / F ]
(b) Describe (in English sentences, as opposed to pseudocode) how `displayFeetAndInches` calculates the number of feet and inches to which the double precision floating point variable `lgth` is equivalent.  

(5 points)

(c) Suppose there exists another subclass of `RoundObject`, called `Sphere`. The `Sphere` class extends the `RoundObject` class, and has methods comparable to that of the `Circle` class. Using UML notation for objects and their inter-relationships, make representations of the different classes and diagram the relationships between them. 

(10 points)
(d) You are asked to describe the following methods in pseudo-code (try to keep this as close as possible to Java syntax).

(i) Describe an instance method in the Sphere class called calcRadius. The method should calculate the radius from the volume information and set the value of the radius variable in the Sphere instance. The volume of a sphere is given by the formula \(\frac{4}{3}\pi r^3\).

(5 points)

(ii) Modify displayFeetAndInches so that it is more sensitive to singular and plural values. For example, it should know when to use “feet” and when to use “foot”. HINT: Use conditional statements.

(10 points)
Describe a class method in the RoundObjects class called `displayMilesFeetAndInches`. The method should take as argument a double precision variable `lgth` that gives some distance in miles, then displays this input in miles, feet and inches. Note that there are 5280 feet to a mile.

(10 points)
(e) Harry, Ron and Hermione come across an old dragon’s lair with numerous marble discs (with etchings indicating their respective top surface areas in square feet) and crystal spheres (with, what do you know(?!?), etchings indicating their respective volumes in cubic feet). They are interested in comparing any two objects and finding out which has the larger radius.

Suppose a Sphere class has been defined that not only extends the RoundObject class but also implements the interface shown in Appendix C.

Describe, in pseudo code (again, as Java-like as possible), a Demonstrate class that would take two RoundObject instances (one Circle and one Sphere) and compare their radii. The comparison should tell you which has a bigger radius, and also give you the ratio of their radii. Feel free to use all methods in the RoundObject, Circle and Sphere classes.

(10 points)
Suppose that you work for the company HPAnalysts (HP as in Harry Potter, not the usual suspect). You are interested in figuring out which two Hogwarts' houses are most popular with the fans. But first, you wish to find out how many such choices there are in total, i.e., how many ways there are to pick two out of four houses.

(a) You know that there are a total of $^4C_2$ (four choose two) ways to select two out of the four houses. And that $^nC_r = n! / [ (n - r)! r!]$, where $n! = n(n-1)(n-2)...(1)$. In pseudocode, write a class method called factorial that calculates $n!$ using a loop. In another class method called choose, use the factorial method to get the value of $^nC_r$. 

(15 points)
(b)  (i) Explain why \( \binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r} \). What is the value of \( \binom{n}{r} \) when \( r=0 \)? What is its value when \( n=r \)?

(5 points)

(ii) Now create a recursive version of the choose method, call it \texttt{recursiveChoose}. Again, describe your method using Java-like pseudocode.

(10 points)
Appendix A

The Sorting Game

Oh, you may not think I'm pretty,
But don't judge on what you see,
I'll eat myself if you can find
A smarter hat than me.
You can keep your bowlers black,
Your top hats sleek and tall,
For I'm the Hogwarts Sorting Hat
And I can cap them all.
There's nothing hidden in your head
The Sorting Hat can't see,
So try me on and I will tell you
Where you ought to be.

You might belong in **Gryffindor**,  
Where dwell the brave at heart,  
Their daring, nerve and chivalry  
Set Gryffindors apart;

You might belong in **Hufflepuff**,  
Where they are just and loyal,  
Their patient Hufflepuffs are true  
And unafraid of toil;

Or yet in wise old **Ravenclaw**,  
If you've a ready mind,  
Where those of wit and learning  
Will always find their kind;

Or perhaps in **Slytherin**,  
You'll make your real friends,  
Those cunning folk use any means  
To achieve their ends.

So put me on! Don't be afraid!  
And don't get in a flap!  
You're in safe hands (though I have none)  
For I'm a Thinking Cap!
public abstract class RoundObject {
    public double radius;
    public RoundObject () { radius = 0.0; }
    public RoundObject (double radius) { this.radius = radius; }

    public void setRadius (double radius) { this.radius = radius; }
    public double getRadius () { return radius; }

    public static void displayFeetAndInches (double lgth) {
        int feet = (int)Math.floor(lgth);
        int inches = (int)Math.round(12.0*(lgth - feet));
        System.out.println(feet + " feet and " + inches + " inches.");
    }

    public abstract void calcRadius ();
    // calcRadius should both calculate and set the value of radius

    public abstract void displayObject ();
    // displayObject should display attributes of the object
}

public class Circle extends RoundObject {
    public double area;

    public Circle () { area = 0.0; }
    public Circle (double area) { this.area = area; }
    public Circle (double radius, double area) {
        super(radius); this.area = area; }

    public void setArea (double area) { this.area = area; }
    public void calcRadius () { this.radius = Math.sqrt(area/Math.PI); }

    public void displayObject () {
        System.out.println("A Circle of Area " + area + " has");
        System.out.println("A Radius that is ");
        displayFeetAndInches(radius);
    }
}

public class Demonstrate {
    public static void main (String argv[]) {
        RoundObject circ = new Circle(100.0);
        circ.calcRadius();
        circ.displayObject();
    }
}
Appendix C

public interface SphereInterface {

    /* Sphere specifies an additional variable, 
       double volume; */

    /* there are three constructors:
       Sphere() - sets radius to 0 and volume to 0
       Sphere(volume) - sets radius to 0 and volume as specified
       Sphere(radius,volume) - sets radius and volume as spec */

    /* a method for setting the volume is required */
    public abstract void setVolume (double volume);

    /* a method for calculating radius from volume info is reqd */
    public abstract void calcRadius ();

    /* a method for displaying the Sphere object is required.
       It assumes that the volume is in cubic feet and displays
       the radius in feet and inches. */
    public abstract void displayObject ();
}