1. What is the output of the following segments of Java code?

(a)  
```java
int x, y, z;
x = 8;
y = 3;
z = x / y;
System.out.println("z is " + z);
```

(b)  
```java
double z = 4 / 7;
System.out.println("z is " + z);
```

(c)  
```java
double x,y,z;
x = 15.0;
y = 5.0;
z = 5 / x+y;
System.out.println("z is " + z);
```

(d)  
```java
System.out.println("4 + 5 = " + 4 + 5);
System.out.println( 4 + 5 + " = 4 + 5");
```

2. Given the function `calc` below:

```java
public double calc(double myVal){
    return myVal * myVal;
}
```

what are the values of the variables `myVal` and `retVal` in the following code snippet after the function has been called?

```java
public void doSomething() {
    int myVal = 4;
    int retVal;

    retVal = calc(myVal);
}
```

3. Write a java statement that converts the radian measure $\frac{\pi}{4}$ to degrees and rounds the result to the nearest degree, using the value 3.1415 for $\pi$. (Hint: see pg. 120 of text for a partial list of Java’s predefined Math functions).
4. Complete the code skeleton below for the CelestialBody class. A CelestialBody contains a 'mass' data member which is initialized by a constructor. In addition to completing the constructor, you will need to write an accessor method to get the mass and a couple of gravAttraction() methods that compute the gravitational attraction between two objects. The formula for gravitational attraction is given as:

\[ F = \frac{G \times m_1 \times m_2}{r^2} \]

where \( G \) is defined as \( 6.67 \times 10^{-11} \) m\(^3\)/(kg\(^s^2\)), \( m_1 \) and \( m_2 \) are the masses of the objects, and \( r \) is the distance between the objects. The first gravAttraction() method takes a mass and a distance as the parameters, whereas the second method takes another CelestialBody and a distance as the parameters. Be sure to define constants appropriately.

class CelestialBody {
    //Constants, etc., should go here.

    // The C.B.'s mass
    private double mass;

    //Complete this constructor:
    CelestialBody(double m) {
    }

    //Provide an accessor method for the mass here.

    //Complete this function:
    public double gravAttraction(double m,  double r) {
    }

    //Complete this function:
    public double gravAttraction(CelestialBody cb, double r) {
    }
}

With the CelestialBody class defined, we can now perform computations such as the following:

public class CelestialBodyUser {

    public static void main(String[] args) {
        CelestialBody earth = new CelestialBody(5.9742e24);
        CelestialBody moon = new CelestialBody(7.34e22);
        double attr;

        attr = earth.gravAttraction(moon, 3.844e8);
        System.out.println("Grav attraction between Earth and Moon is: "+ attr + " Newtons.");
    }
}