1. Fill in each blank with the proper java code.

public class TotalItUp

{

private int numOne, numTwo, answer;

public void setNums(int n1, int n2)  
 {

}

//method add totals up numOne and numTwo

public void add()  
 {

}

//method print displays the sum of numOne and numTwo

public void print()  
 {

}

}

//code in the main of another class

***OUTPUT***

32 + 21 = 53

15 + 8 = 23

TotalItUp test = new TotalItUp();

test.setNums(32,21);

test.add();

test.print();

test.setNums(15,8);

test.add();

test.print();

2. Fill in each blank with the proper java code.

public class CompareNums

{

private int one,two;

public CompareNums(int n1, int n2)  
 {

setNums(n1,n2);

}

public void setNums(int n1, int n2)  
 {

}

//returns true of one == two

//returns false if one != two

public boolean compare()  
 {

}

//returns one and two are equal if ==

//returns one and two are not equal is !=

public String toString()  
 {

}

}

***OUTPUT***

32 and 32 are equal.

15 and 8 are not equal.

**//code in the main of another class**

CompareNums test = new CompareNums(32,32);

System.out.println(test);

test.setNums(15,8);

System.out.println(test);

3. Fill in each blank with the proper java code.

class Table

{

private int upperBound;

private int multiplier;

public Table(int bound, int mult)

{

}

public void setNums(int bound, int mult)

{

}

public void print()

{

***OUTPUT***

1 - 5

2 - 10

3 - 15

4 - 20

5 - 25

6 - 30

7 - 35

8 - 40

9 - 45

10 - 50

}

}

**//code in the main of another class**

Table table = new Table(10,5);

table.print();

4. Extend class Monster to make a Gargoyle. A Gargoyle is a Monster. A Gargoyle has all of the properties and behaviors of a Monster and it has an additional property of type double that stores the agility of the Gargoyle. You must provide 3 constructors for class Gargoyle. You must provide a toString( ) method.

public class Monster

{

private String myName;

public Monster()

{

myName = "Monster";

}

public Monster( String name )

{

myName = name;

}

public String toString()

{

return myName + "\n";

}

}

5. Write a Dot class which implements the Locatable interface. Dot will have x, y, and size properties. x, y, and size are whole numbers. You must provide 3 constructors for class Dot. You must implement the methods listed in the Locatable interface.

public interface Locatable

{

public int getXPos();

public int getYPos();

}

One constructor must be a default.

One constructor must be an x and y only constructor.

One constructor must be an x, y, and size constructor.

You must provide a toString() method.

The toString() should return the x, y, and size of the Dot.