1. Using *inheritance*, one class can acquire the properties of others. Consider the following *Animal* class:

class Animal{

void walk(){

System.out.println("I am walking");

}

}

This class has only one method, *walk*. Next, we want to create a *Bird* class that also has a *fly* method. We do this using *extends* keyword:

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

}

Finally, we can create a Bird object that can both *fly* and *walk*.

public class Solution{

public static void main(String[] args){

Bird bird = new Bird();

bird.walk();

bird.fly();

}

}

The above code will print:

I am walking

I am flying

This means that a Bird object has all the properties that an Animal object has, as well as some additional unique properties.

The code above is provided for you below. You must add a *sing* method to the *Bird* class, then modify the *main* method accordingly so that the code prints the following lines:

I am walking

I am flying

I am singing

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

class Animal{

void walk(){

System.out.println("I am walking");

}

}

class Bird extends Animal{

void fly(){

System.out.println("I am flying");

}

}

public class Solution{

public static void main(String args[]){

Bird bird = new Bird();

bird.walk();

bird.fly();

}

}

Write the following code in eclipse editor below:

1. A class named *Arithmetic* with a method named *add* that takes  integers as parameters and returns an integer denoting their sum.
2. A class named *Adder* that inherits from a superclass named *Arithmetic*.

**Note:** Because multiple classes are being written in the same file, you *must not* use access modifiers (e.g.: ) or your code will not execute.

**Input Format**

You are not responsible for reading any input from stdin; a hidden code checker will test your submission by calling the *add* method on an *Adder* object and passing it  integer parameters.

**Output Format**

You are not responsible for printing anything to stdout. Your *add* method must return the sum of its parameters.

**Sample Input**

We will append the following *Solution* class to your submitted code, which checks for inheritance and passes the necessary arguments to your *add* method:

class Solution{

public static void main(String []args){

// Create a new Adder object

Adder a = new Adder();

// Print the name of the superclass on a new line

System.out.println("My superclass is: " + a.getClass().getSuperclass().getName());

// Print the result of 3 calls to Adder's `add(int,int)` method as 3 space-separated integers:

System.out.print(a.add(10,32) + " " + a.add(10,3) + " " + a.add(10,10) + "\n");

}

}

You *do not* need to write a *Solution* class.

**Sample Output**

The *main* method in the *Solution* class above should print the following:

My superclass is: Arithmetic

42 13 20