Homework #1:

import java.util.Scanner;

public class addtwonumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner( System.in );

System.out.println("This program adds two numbers togethers, please enter the first number");

int m = scanner.nextInt();

System.out.println("Please enter the second number");

int n = scanner.nextInt();

int sum = m + n;

System.out.println("The answer is "+sum);

}

}

Homework #2:

What is the difference between Class and Object? Use UML to depict Class and Object.

In object-oriented software, real world objects migrate into the code. In programming terms, our objects become stand-alone modules with their own knowledge and behavior or, if you prefer, their own data and processes. It’s common to think of a software object as a robot, an animal, or a little person: each object has certain knowledge, in the form of attributes, and it knows how to perform certain operationsfor the benefit of the rest of the program. For example, a person object might know its title, first name, last name, date of birth and address; it would be able to change its name, move to a new address, and tell us how old it is, and so on.

A class encapsulates characteristics common to a group of objects. There are a number of ways you could think of a class:

A factory manufactures objects according to some blueprint.

A set specifies what features its member objects will have.

A template allows us to produce any number of objects of a given shape.

A dictionary definition describes an object as precisely as possible.

In UML, classes are drawn as boxes on a class diagram. So that we can easily tell the difference between classes and objects, class names (on class diagrams) are not underlined, while object names (on object diagrams) are. Classes and objects are rarely mixed on the same diagram: it turns out that we can do most of our modeling in terms of classes, reserving object diagrams for illustration and verification purposes. Object-oriented programmers are often heard to say ‘Every object is an instance of a class’, hence the use of the term instance as a synonym for object.