**1. Write the output for each section of print statements. (10 points)**

box(l,w) = (10,10)

box(l,w) = (10,10)

box(l,w) = (10,10)

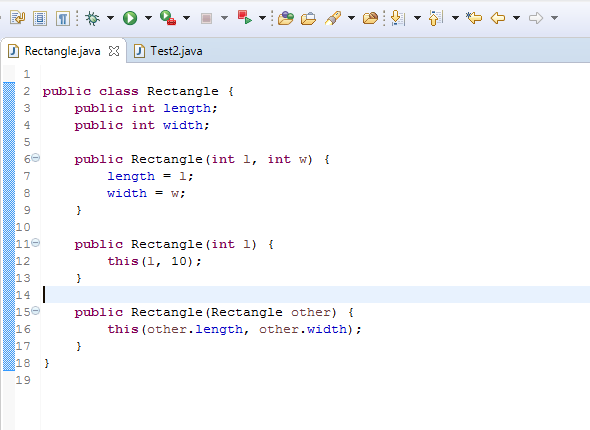
a = 0

a = 100

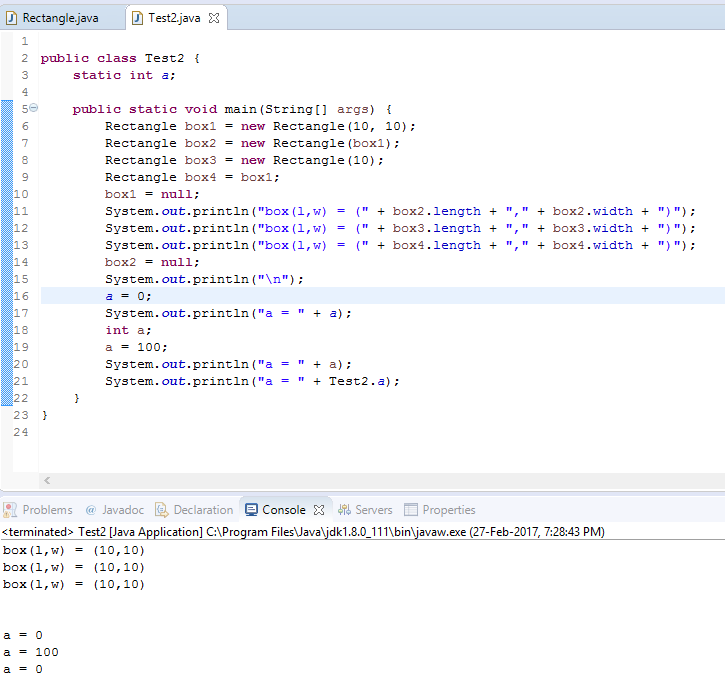
a = 0

Please find below the screenshots of the program:

Screenshot 1:



Screenshot 2:



**2. Which Rectangle constructor does instance box1 call? Why? What concept does this demonstrate? (10 points)**

The box1 instance calls the **Rectangle (int l, int w)** constructor because the box1 is instantiated with **new Rectangle (10, 10),** the parameters and datatype in the above declaration are matched with **Rectangle (int l, int w)**. The concept here demonstrates constructor overloading.

Construction Overloading:

Constructor in java is a special type of method that is used to initialize the object.

Java constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.

There are two types of constructors:

Default constructor (no-arg constructor)

Parameterized constructor

**3. Which Rectangle constructor does instance box2 call? Why? What concept does this demonstrate? (10 points)**

The box2 instance calls the **Rectangle (Rectangle other)** constructor because the box2 is instantiated the **Rectangle (box1),** and this inturn calls the constructor **Rectangle (int l, int w)**. The constructor **Rectangle (box1)** is called here because we are passing the instance of Rectangle ie: box1, this matches the constructor **Rectangle (Rectangle other)**. The concept here demonstrates constructor overloading. Calling a constructor from the another constructor of same class is known as Constructor chaining.

**4. Which Rectangle constructor does instance box3 call? Why? What concept does this demonstrate? (10 points)**

The box3 instance calls the **Rectangle (int l)** constructor because the box3 is instantiated with **Rectangle (10)** constructor, in this constructor this intern calls the constructor **Rectangle (int l, int w)**. The constructor **Rectangle (int l)** here because we are passing the parameter that matches the **Rectangle (int l)** constructor. The concept here is constructor overloading.

**5. What is the connection between box1 and box4? What happens (memory wise) when both box1 and box 4 are declared null? (20 points)**

The connection between box1 and box4 here is the box4 holds the memory address of box1 it is done with the following statement Rectangle box4 = box1;. If both box1 and box4 are declared null they points to memory location of nothing “null” means it doesn’t point to any memory location. And if box4 = null is assigned after print statement it doesn’t not show any effect in our program, because we are using box4 only in print statement however once the main method is executed and the control comes out of it they garbage collector comes into the picture and clear all the memory allocated in the main method.

**6. In the Test2 class, there are two variables named a. What is the difference between the two variables? What concept does the output of the last three print statements demonstrate? Explain. (20 points)**

The difference between two variables is one is static variable (class variable) and one is instance variable.

Java static variable is a variable which belongs to the class and not to object(instance)

Static variables are initialized only once , at the start of the execution . These variables will be initialized first, before the initialization of any instance variables.

A single copy to be shared by all instances of the class

A static variable can be accessed directly by the class name and doesn’t need any object

Syntax : <class-name>.<variable-name>

where as

the instance variable create the new memory whenever new object is created.

The last three statements show the concept of shadowing, here there are two variables one is static and one is instance variable with the same name "a" in this the highest level scope variable is hidden by lower level because the lower level variable overrides the higher one, but here the last print statement the static variable is accessed by class name.

**7. In the Rectangle class, explain the implementation of the Rectangle (int l) constructor. (10 points)**

The constructor Rectangle (int l) implementation we see the key word “**this**” the this keyword is used to refer the current object’s instance method or constructor, supporting the above statement we have created the instance of Rectangle box3 = new Rectangle (10); the second statement of constructor Rectangle (int l) is the “**this (l ,10)**”, so here box3 is the object of Rectangle class and the statement this(l, 10) refers to the instance constructors of object box3 so this invokes Rectangle(int l, int w) this concept is called explicit constructor invocation.

**8. In the Rectangle class, explain the implementation of the Rectangle (Rectangle other) constructor. (10 points)**

The implementation of Rectangle (Rectangle other) is similar to the above question the second statement in the constructor Rectangle (Rectangle other) is this (other.length, other.width); as the this keyword refers to the instance methods and constructors of current object, box2 is the object of Rectangle class and the this keyword searches for the constructors that matches the arguments and it calls the Rectangle (int l, int w), here we called the other constructor of the same class from one constructor using “**this**” keyword, this is called explicit constructor invocation.