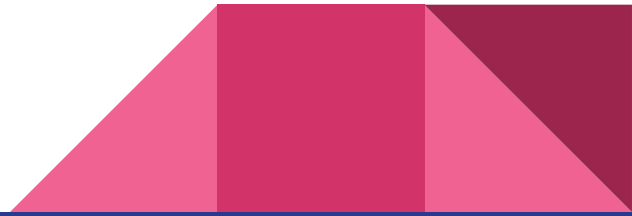
Objects - Classes

Part 2

RECAP: Using the Constructor

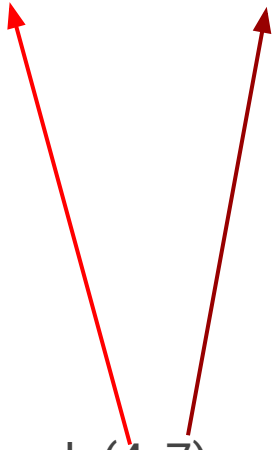
Rectangle rect = new Rectangle(4, 6)

In order to create an object we must use the new operator to instantiate a new class object. Instantiation is the process of creating an instance of an object which allocates memory for a new object and references that object in memory.



RECAP:Using the Constructor

```
public Rectangle(int myWidth, int myHeight){
    width = myWidth;
    height = myHeight;
}
```



The arguments are passed using call by value into the parameters.

Rectangle rect = new Rectangle(4, 7);

Arguments parameters must match the types identified in the parameters.

Creating Multiple Constructors

Let's say we want to create squares. We can actually write:

```
Rectangle rect = new Rectangle(16);
```

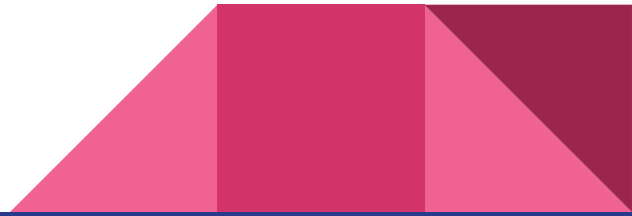
We can add additional constructors that take different number of parameters.

```
public Rectangle(int size){
        width = size;
        height = size;
}
```

Constructors do not need to have parameters in order to create an object

Having multiple constructors with the same name but different parameters is called **overloading**.

The compiler knows which constructor to use.

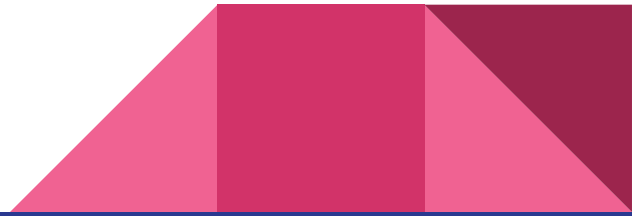


Constructor: No-argument (default constructor)

No-argument constructor set objects to a default value.

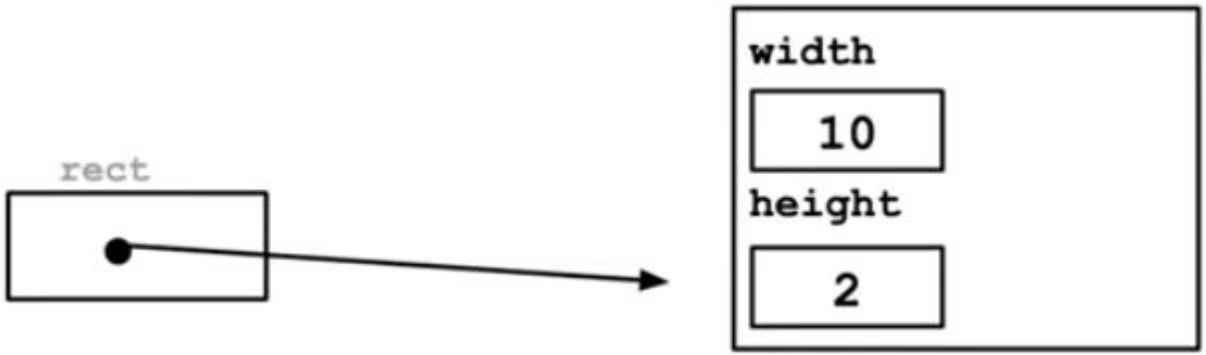
```
public Rectangle(){

}
```

```
Rectangle rect = new  Rectangle();
```



Objects in Memory

In memory the variables simply stores a location or a reference



Objects in Memory

When we write:

Rectangle rect;

The variable is not pointing at any object data. When an object reference is not pointing to any object data, it is considered to be **null.** Null objects references do not allocate any memory.

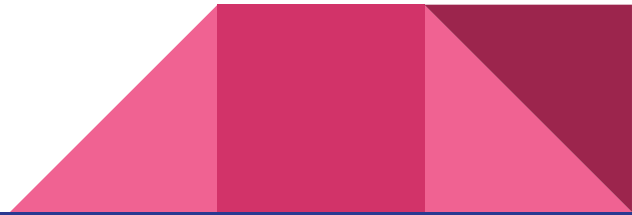
Static and Non-static methods

A **static method** is a method that belongs to a class, but it does not belong to an instance of that class and this method can be called without the instance or object of that class. They may not use non-static methods. Example: static methods in class “Math.abs()”, “Math.pow()”, “Math.PI”

```
public static void printMsg(){
  System.out.println("Hi! Rectangle class");
}
```

Non-static belongs to each object that is generated from the class. Methods can access any **static** method and **static** variable.

```
public int calcArea(){
  return width * height;
}
```



Exercise

Create a class Employee

Define instance variables

Define at least 3 construtor (1 of them default constructor)

Declare static and non-static methods

Create multiple objects from your Driver and test the constructors and methods.