3. Objects and Classes :: Constructors

All classes must have at least one constructor even if that constructor does nothing:

```
public class C {
  public C() {
  }
}
```

The **default constructor** of a class is the constructor that has no parameters. If you do not write a constructor, the compiler automatically generates a default constructor as shown above.

The job of a constructor is to initialize the object by initializing one or more of the instance variables of the object.

Constructors may be **overloaded**.

```
public class Point {
  public Point() {  // Default constructor.
    setX(0);
    setY(0);
  }
  public Point(double initX, double initY) {  // A second constructor.
    setX(initX);
    setY(initY);
  }
}
```

3. Objects and Classes :: Automatic Instance Data Initialization

Any instance variables that are not initialized in a constructor will be automatically initialized. **int** and **double** data members are initialized to 0 and objects are initialized to **null**.

```
public class C {
  private int x;
  private double y;
  private String s;
  public C() {
  }
}
public class Main() {
  public static void main(String[] args) {
    C cObject = new C();
  }
}
```

3. Objects and Classes :: References

When we declare an object variable it is initialized to **null**. When the object is instantiated the object variable will be initialized with a **reference** to the object.

```
Point pete;
pete = new Point(10, 20);
```

Assigning object variable to another causes both object variables to refer to the same object.

```
Point patty = pete;
```