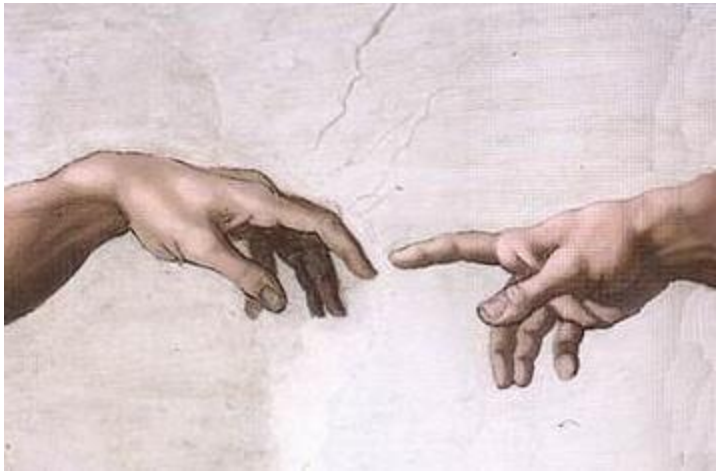
Anatomy of an Object



Anatomy of an Object

- **Instance Variables**
- **Methods**
  - **Constructors** – Initialize our objects. They are invoked whenever an object is instantiated (created) with the keyword **new**.
  - **Accessor methods** – allow access to private data, but do not actually change the data.
  - **Mutator methods** – These do change the data… we say they change the 'state' of the object.
  - **toString()** – To String is a special method that is used to return the state of the object as a String. It's a good idea to include the Instance Variables and the result of any key method calls here.

Instance Variables & Encapsulation

We keep our Instance Variables private so that a client can only interact with our data via the public interface (methods) for that object.

Ex:


```
public class Triangle
{
        private int base, height;
        …
}
```

What does a constructor do?

- The purpose of a constructor is to initialize the STATE of the object.
- The state of an object is the value(s) of it's data (Instance Variables).
- Constructors give initial values to an objects data and instantiate any of it's objects.

What makes **constructors** unique?

- They have NO return type
- They share the name of the class
- They are **invoked** implicitly whenever someone **instantiates** (creates) an object with the keyword **new.**

Ex's:

```
public Die() = new Die();
public Car() = new Car("Ford");
public Monster() = new Monster("Ed");
```

Default / No Arg Constructor

- When we build a class, we generally include a default, no argument constructor.
- The default constructor provides reasonable default values for the Instance Variables.

```
public class Triangle
{
    private double base, height;
    public Triangle()
    {
        base = 0.0;
        height = 0.0;
    }
}
```

```
public class TriangleRunner
{
    Triangle t = new Triangle();

    …
}
```

Initialization Constructor

- A constructor with parameters.
- Allows the client to provide their own initial values at the time an object is instantiated (created) with the keyword new.

```
public class Triangle
{
    private double base, height;
    public Triangle(double b, double h)
    {
        base = b;
        height = h;
    }
}
```

```
public class TriangleRunner
{
    Triangle t = new Triangle(4, 6);
    …
}
```

String toString()

- In Java, toString() is an important method to include in your objects.
- toString() returns a String that represents the current 'state' of the object. All of it's data and any key method calls should be included.
- toString() gets invoked implicitly (automatically) whenever an object is printed out.

```
public class Triangle
{
    private double height, width;
    …
    public String toString()
    {
            String result="\nBase: " + base;
            result += "\nHeight:" + height;
            result +="\nArea:" + getArea();
            return result;
    }
}
```

```
public class TriangleRunner
{
    Triangle t = new Triangle(5,10);
    System.out.println(t);
}
```