

AP Computer Science
8.03 Collaboration
Mrs. Draeger

**READY.
SET. CODE.**

Instructions:

**Work with your team to write a class with a default constructor and methods.
You can choose a book, candy, or phone as your object.**



Team Members:

Toni-Ann

Luis Vega

```
public class Phone {  
    // object class used to write constructor and methods  
  
    // default constructor  
    public Phone()  
    {  
    }  
  
    // methods  
    public void ringtone()  
    {  
        System.out.println("... a ringtone plays...");  
    }  
}
```

```

    public void system(string system)
    {
        System.out.print("System: " + system);
    }
    Public void type(String type)
    {
        System.out.println("Type of phone: " + type);
    }
    public void color(string color)
    {
        System.out.print("Color" + color);
    }

```

// main method used to create variables, objects and call methods

```

    public static void main(String [] args)
    {
        Phone example = new Phone();
        example.ringtone();
        example.type("iPhone");
        example.system("IOS");
        example.color("red");

    }

```

} // end of class

----- **** ----- **** ----- **** ----- **** ----- **** ----- **** -----

Team Members:

Taylor Wassell

Akil Mohideen

// object class used to write constructor and methods

```
public class Candy {
```

```
// default constructor
```

```
public Candy()
```

```
{  
}
```

```
//Methods
```

```
public void name(String name)
```

```
{  
  
}
```

```
    System.out.println("Name: " + name)
```

```
public void taste(String taste);
```

```
{  
  
}
```

```
    System.out.println("Taste: " + taste);
```

```
public void type (String type)
```

```
{  
  
}
```

```
    System.out.println("Type (i.e. hard, chewy, soft): " + type);
```

```
public void color(String color)
```

```
{  
  
}
```

```
    System.out.println("Color(s): " + color);
```

```
// main method used to create variables, objects and call methods
```

```
public static void main (String [] args)
```

```
{
```

```
Candy Yum = new Candy();  
Yum.name("Skittles");  
Yum.taste("sweet");  
Yum.type("chewy");  
Yum.color("Multi-colored");  
}
```

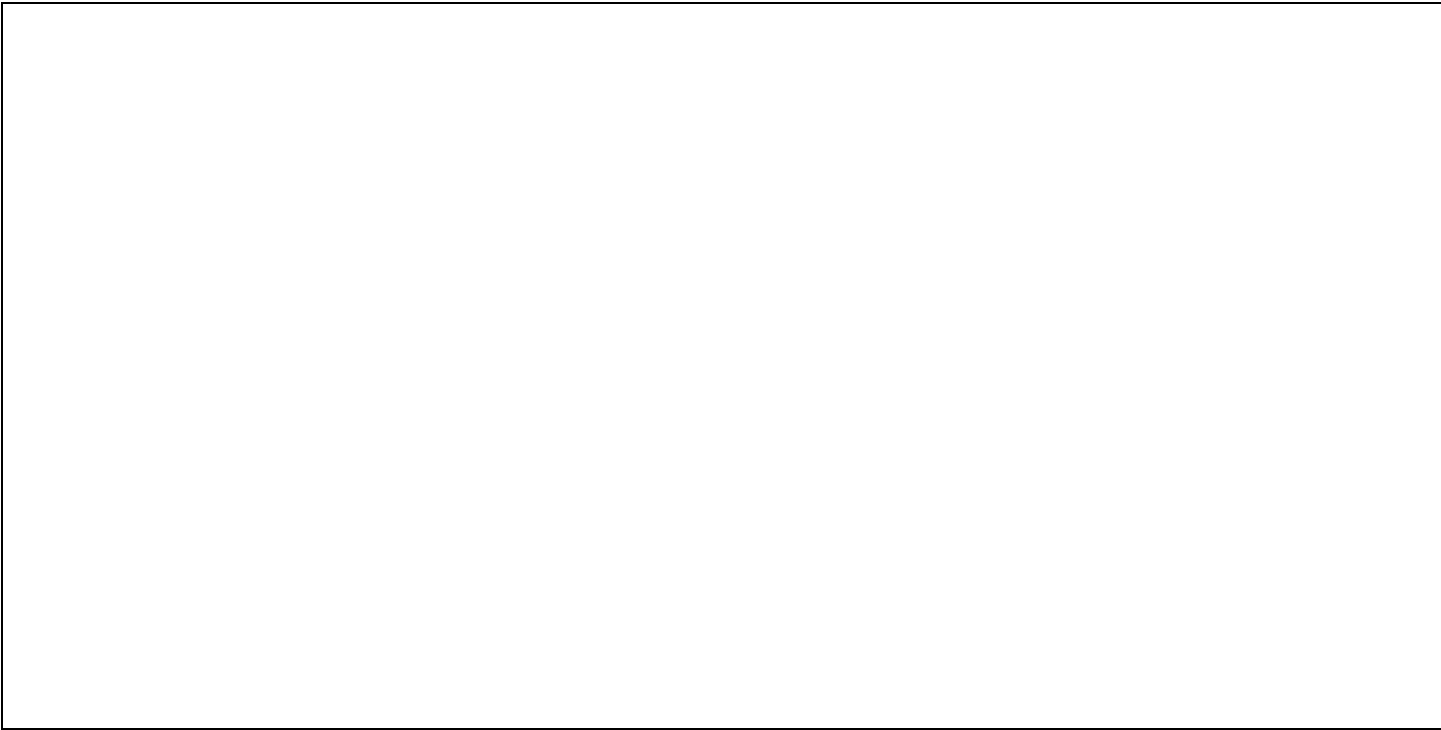
```
// end of class
```

```
}  
}
```

Take-away notes:



When using the main method, the class is first named and then initialized using `Class object = new Class()`. That object is then used with the dot method to use all of the methods already defined.



----- ***** ----- ***** ----- ***** ----- ***** ----- ***** ----- *****
----- ***** ----- *****

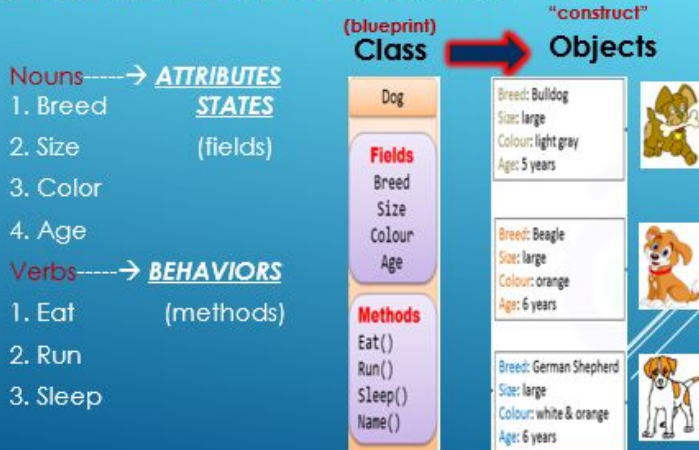
SLIDES FROM LESSON

----- ***** ----- ***** ----- ***** ----- ***** ----- ***** ----- *****
----- ***** ----- *****

Vocabulary

1. Class - a blueprint for an object
2. Object - created from a class
3. Constructor - used to initialize a new object
4. Method - Statements to perform an operation

EXAMPLE: CLASSES AND OBJECTS



```
public class Dog
{
    public Dog()
    {
    }

    public void sound()
    {
        System.out.println("Bark!");
    }

    public void breed(String type)
    {
        System.out.println("Breed: " + type);
    }

    public void name(String dogName)
    {
        System.out.println("Sit " + dogName + " sit! Good dog!");
    }

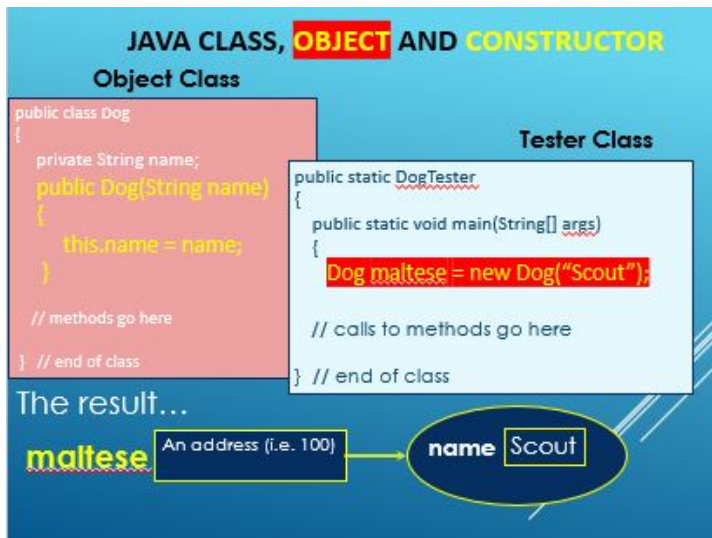
    public static void main(String[] args) {
        Dog friend = new Dog();
        friend.breed("Labradoodle");
        friend.name("Amber");
        friend.sound();
    }
}
```

DOG CLASS

- **Constructor**
- **Methods**
- **Main method**
- **Create object**
- **Method calls**



Looking ahead to a 2-class implementation...



Constructors vs. Methods

- A constructor in Java is a block of code similar to a method that's called when an instance of an object is created. Here are the key differences between a constructor and a method:
- A constructor doesn't have a return type.
- The name of the constructor must be the same as the name of the class.
- Unlike methods, constructors are not considered members of a class.
- When a new instance of an object is created, the object is assigned any private instance variables.

