
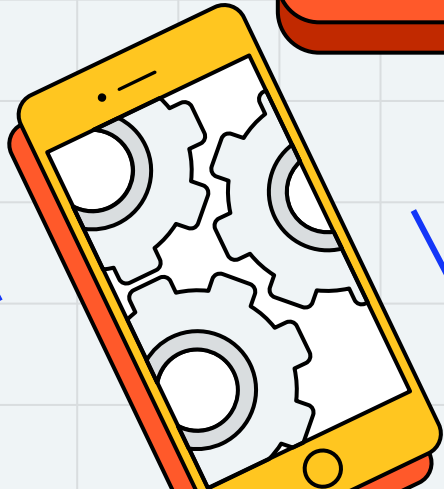




Application

Programming



Hend Alkittawi

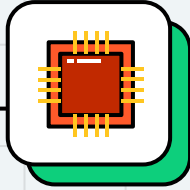




Java Basics

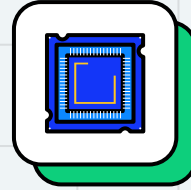
Introduction to Java Concepts and
User-defined Java Classes

CONCEPTS



JAVA

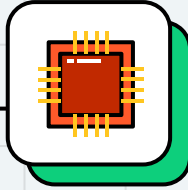
- Released in early 1990s
- Object-oriented
- Java SE, Java EE, Java ME



OBJECT

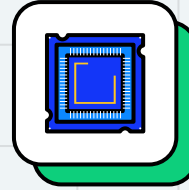
- Defined by a **class**
- A class is a reusable software components
- Non-primitive data type

CONCEPTS



JAVA APIs

- Application Programming Interfaces
- Collections of existing classes and methods in the Java class libraries

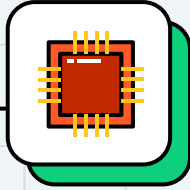


JDK

- Java Development Kit
- Includes compiler, loader, debugger, **JRE**, and other tools for Java development

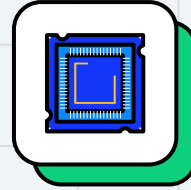


CONCEPTS



JVM

- Java **V**irtual **M**achine
- A part of the JRE that executes bytecodes.
- Bytecodes are platform independent (**portable**)



IDE

- **I**ntegrated **D**evelopment **E**nvironment
- A software that provides comprehensive facilities for software development.



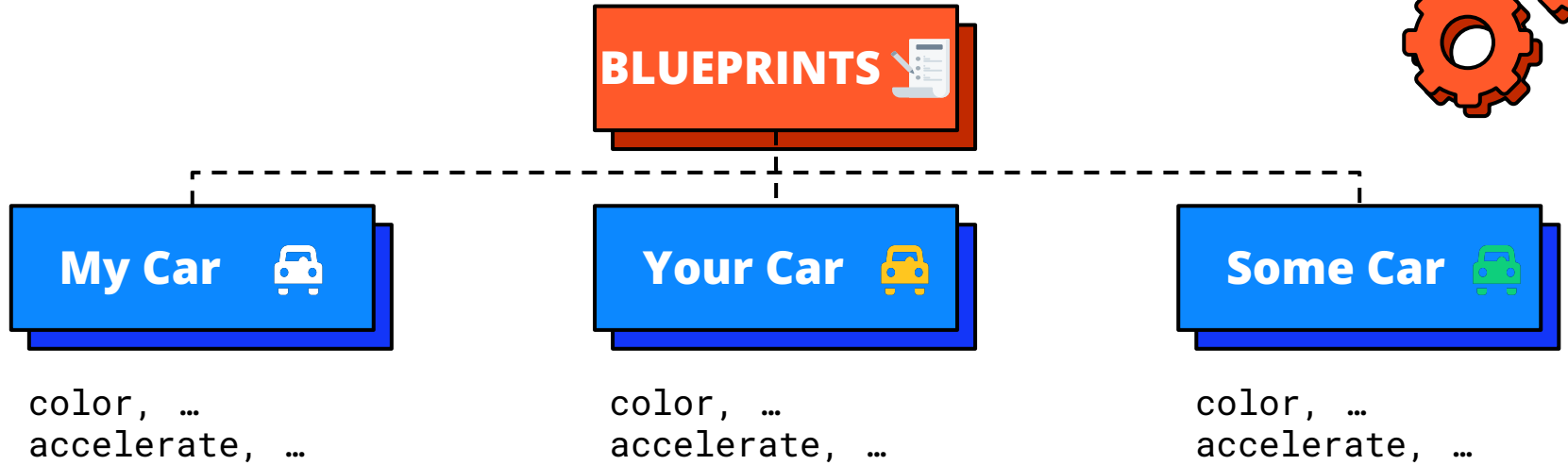
WHY DO WE USE OBJECTS?

- Let us discuss the difference between
 - Write a program to read n number of values in an array and display them in reverse order.
 - Write a program to calculate x raised to the power n (x^n).
 - Write a program to swap two numbers.

and

- Small Scale Applications

OBJECT ANALOGY



JAVA OBJECTS

```
public class Car {
```

Class Declaration

```
private int colorID;  
private double speed;
```

Attributes / Instance Variable(s)

```
public void setSpeed(int s){  
    speed = s;  
}  
public void accelerate(){  
    speed = speed + 70;  
}
```

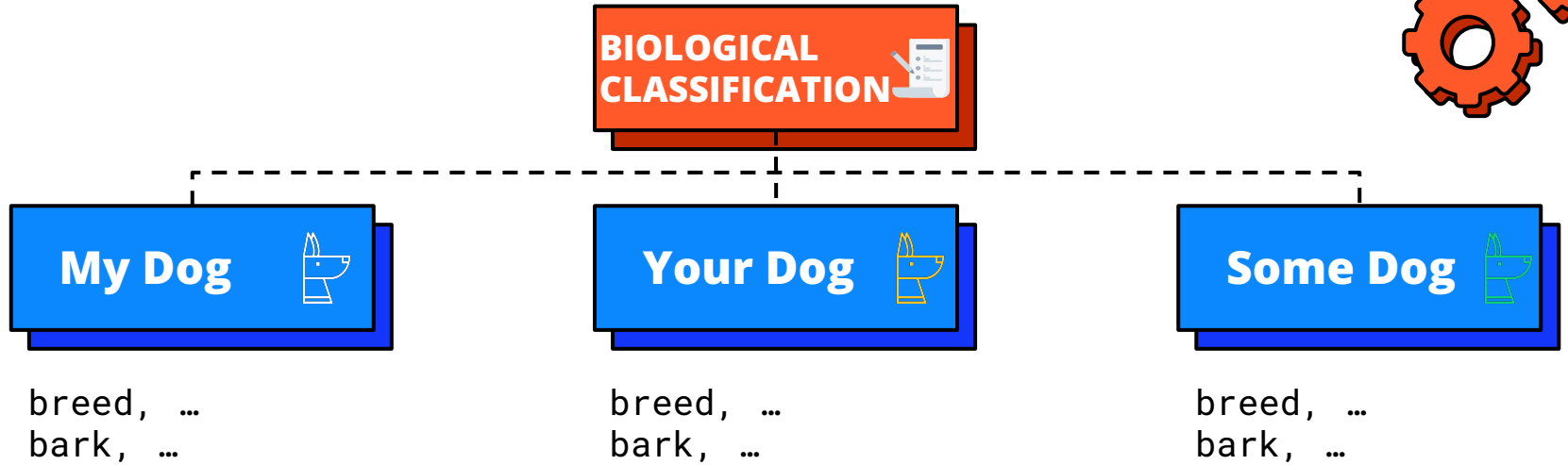
Behaviors / Method(s)

```
public static void main(String[] args){  
    Car myCar = new Car();  
    myCar.setSpeed(50);  
}
```

Instantiation (building a Car object)

Car.java

OBJECT ANALOGY



JAVA OBJECTS

```
public class Dog {
```

Class Declaration

```
private int breedID;
```

Attributes / Instance Variable(s)

```
public void bark(){  
    System.out.println( " Woof " );  
}
```

Behaviors / Method(s)

```
public static void main(String[] args){  
    Dog myDog = new Dog();  
    myDog.bark();  
}
```

Instantiation (building a Dog obj)

Dog.java

THE String CLASS

- The `char` primitive type stores one character of the Unicode character set.
- A string is a sequences of characters.
 - can be 0 or more characters long
 - "" is the empty string
- The `String` class facilitates handling multiple characters at once.
- A `String` object can be declared in order to work with strings

```
String someStringObject = new String();  
someStringObject = "HI";
```

THE String CLASS

- Many String manipulation methods are available, here are some:
 - `split()` method to get an array of Strings from a String, based on a delimiter. This object method takes a delimiter as a parameter.
 - `trim()` method to clear off any additional space from the text. This object method takes no parameters.
 - `charAt()` method to get the character at an index in the string [starting with index 0]. This object method takes an index as a parameter.
 - `equals()` method to check if two Strings contain the same text. This object method takes a String object as a parameter.

THE String CLASS

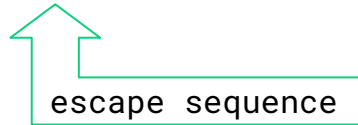
- Java also treats *string literals* as objects
 - A string literal is a quoted string: "Sam I am", "Hi", ..., etc.
- Examples:

```
String strObj = "This is \"a\" String";
```

```
strObj.charAt(0);
```

```
"HI ".trim();
```

```
"Hello".equals(strObj);
```



- Note how we can create/work with string literals as String objects.

JAVA OBJECTS

```
public class Dog {  
  
    private int breedID;  
    private String name;  
  
    public void bark(String dogBark){  
        System.out.println(dogBark);  
    }  
  
    public static void main(String[] args){  
        Dog myDog = new Dog();  
        myDog.bark( " Woof " );  
    }  
}
```



Dog.java

Class Activity

```
public class SimpleLocation {  
  
    private double latitude;  
    private double longitude;  
  
    public void setLocation(double lat, double lon){  
        latitude = lat;  
        longitude = lon;  
    }  
  
    public static void main(String[] args){  
        SimpleLocation utsa = new SimpleLocation();  
        utsa.setLocation(32.9, -117.2);  
    }  
}
```

What is
the value
of
longitude
?



THANK

YOU!



DO YOU HAVE ANY QUESTIONS?



hend.alkittawi@utsa.edu



By Appointment



Online