Java

Today

- Objects
- Data Types / Collections
- Iteration
- Selection
- Operators
-

Java's Big Idea

Virtual Machine with C style notation

Write Once Run Anywhere

Five Primary Goals of the Java Language

1. It should be "simple, object-oriented and familiar"
2. It should be "robust and secure"
3. It should be "architecture-neutral and portable"
4. It should execute with "high performance"
5. It should be "interpreted, threaded, and dynamic"

Concurrent and threaded from the beginning

By the Numbers

- 930 million JRE Downloads every year
- 3 billion Mobile Phones run Java
- 9 million Java Developers in the world
- #2 (behind JavaScript) Github new Repos created with 283354

Code!!!!

```java
// Hello World

package com.flatironschool;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Primitive Data Types

```
//All primitives have Object Wrappers

float x = 20.25; //So if I need a float 'Object' I would use Float x = 20.25;

boolean x = true; // Or Boolean x = true;

char x = ‘x’; // Or Character x = 'x';

byte x = 0x03; // Or Byte x = 0x03;

short x = 15; // Or Short x = 15;

int x = 20; // Or Integer x = 20;

long x = 9,274,387,302 // Or Long x = 9,274,387,302;
```

Data Structures

```
List<String>myList = new ArrayList<String>(); //List<t>
Map<String, String>myMap = new HashMap<String, String>(); //HashMap<t,t>
String[] myArray = new String[10]; //String array with a capacity of 10
String[] myArray = {"string1", "string2", "string3"}; //String array with literal
```

Variables

```
String name;
name = "Al Tyus";

String name = “Al Tyus”;

int num = 20;
```

Operators

- **Assignment:** =, +=, -=, /=, %=, etc…
- **Additive:** +, -
- **Multiplicative:** *, /, %
- **Relational:** <, >, <=, >=
- **Equality:** ==, !=
- **Logical AND:** &&
- **Logical OR:** ||

Control Flow

if-then

```
int x = 15;

if (x > 10) {
    System.out.println(x + “ is greater than 10”);
}
```

if-then-else

```
int x = 15;

if (x > 10) {
    System.out.println(x + “ is greater than 10”);
}
else if (x < 5) {
    System.out.println(x + “ is less than 5”);
}
else {
    System.out.println(x + “ is neither greater than 10 or less than 5”);
}
```

Iteration

```
for (int i = 0; i < 100; i++){} // for loop

for (Integer i : myArray){} //foreach loop

while (true){} //while loop

do {}while(true) //do while loop
```

Class

```
public class Person {
    //class body
}
```

Methods

Instance Method

```
private void grow(int inches) {
    //can access both static and instance variables and methods
}
```

Static Method

```
    private static Time currentTime(){
    return Time.now(); //Only have access to static variables and methods
}
```

Constructor

```
    public class Person {
        private String mName;

        public Person(String name){

        mName = name;
    }
}
```

Getters and Setters

```
public class Person {
    private String mName;

    public Person(String name){
        mName = name;
    }

    public String getName(){
        return mName;
    }

    public void setName(String name){
        mName = name;
    }
}
```

Inheritance

```
public class Main {
    public static void main(String[] args) {
        MountainBike mb = new MountainBike();
        System.out.println(mb.getColor());  //prints red
    }
    public static class Bike{
        private static String mColor;
        public Bike(){
            mColor = "red";
        }
        public String getColor(){
            return mColor;
        }
    }
    public static class MountainBike extends Bike{}
}
```

Interfaces

Declaring Interfaces

```
public interface Animal {
    public void eat();
    public void move();
}
```

Implementing Interfaces

```
public class Cat implements Animal {
    public void eat(){
    }
    public void move(){
    }
}
```