Introduction to JAVA

MNNIT Computer Coding Club

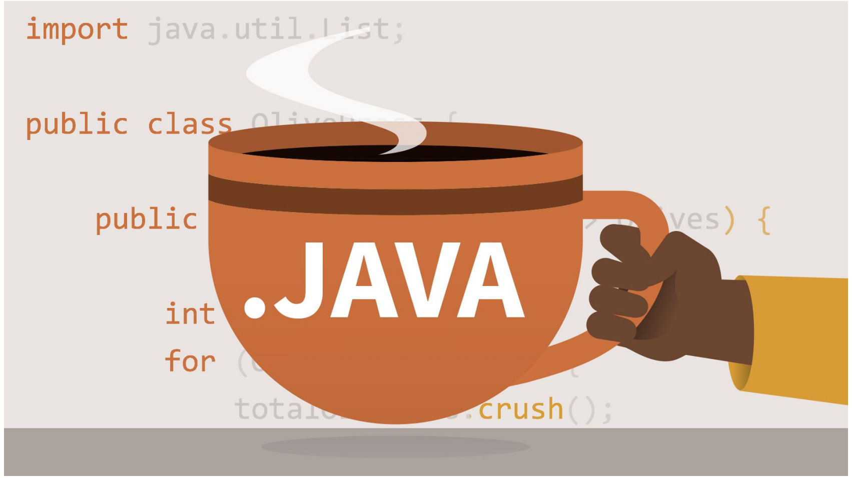
Objective

To make you answer the following questions yourselves:

1. Why to learn Java?
2. What is OOP?
3. How to write your first Java program?
4. What are classes and objects?
5. What more is in Java?

Reference

1. Java: The Complete Reference by Herbert Schildt
2. Head First Java by Bert Bates and Kathy Sierra
3. Oracle: Java documentation
4. Lots of online tutorial ;)



Introduction to Java

According to Wikipedia, Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It first appeared in May 23, 1995 and is designed by James Gosling.

Java is one of the most popular and widely used programming languages and platforms. (A platform is an environment that helps to develop and run programs written in any programming language.)

History of Java

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of Java starts with the Green Team.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic". Java was developed by James Gosling, who is known as the father of Java, in 1995. James Gosling and his team members started the project in the early '90s.

Home Assignment

1. Why Java is named so?
2. What is Oak and Greentalk in reference to Java?

Features of Java

1. Java is Platform Independent.
2. Java is Object oriented.
3. Java is secure.
4. Java is Robust.
5. Java is Portable.
6. Java is multithreaded.
7. Java is Interpreted.

And lot more...

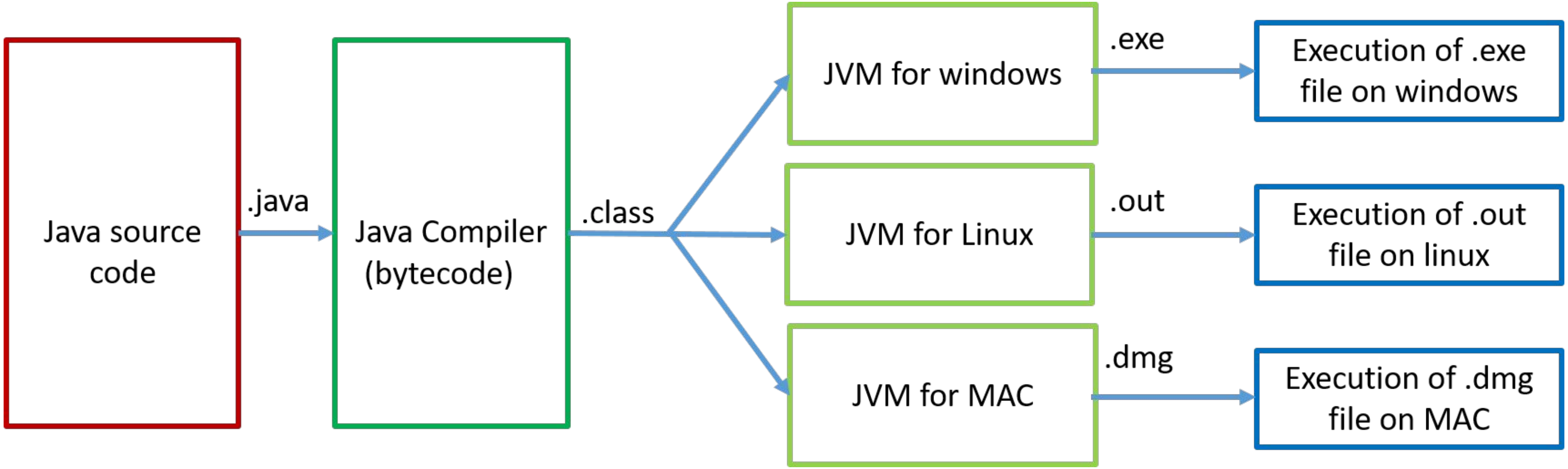
Java Setup - Progress

Status: Installing Java

Platform Independent

Platform independence is a term that describes a technology (usually a Programming Language or a FrameWork) that you can use to implement things on one machine and use them on another machine without (or with minimal) changes.

Platform independence



Robust

Robustness is the capacity of a computer system to handle the errors during execution and manage the incorrect input of data. Java is robust because it utilizes strong memory management. There is an absence of pointers that bypasses security dilemmas.

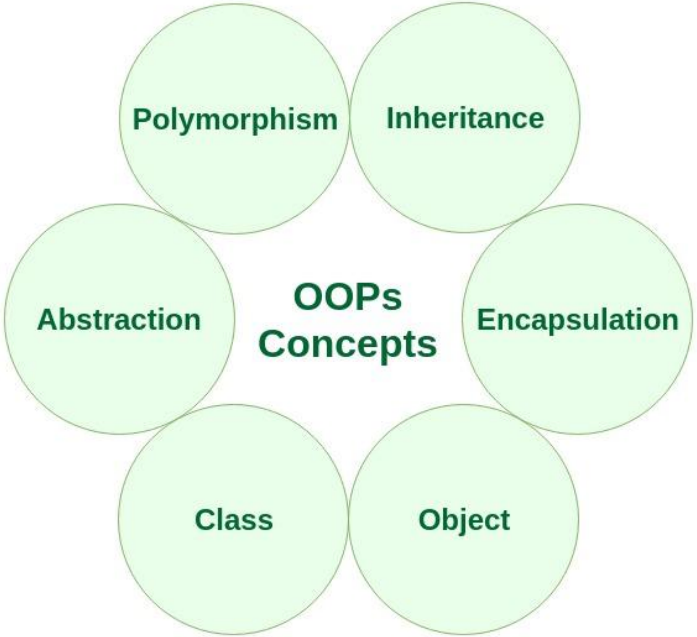
There is automatic garbage collection in Java which runs on the Java Virtual Machine to eliminate objects which are not being accepted by a Java application anymore. There are type-checking mechanisms and exception-handling in Java. All these features make Java robust.

Object Oriented Programming (OOP)

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects.

Apart from OOP the other popular programming paradigm is Procedural programming.

Procedural Oriented Programming is a programming language that follows a step-by-step approach to break down a task into a collection of variables and routines (or subroutines) through a sequence of instructions.



Home Assignment

Can you find out the name of first programming language that follows OOP?

Class

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

Class defines a new data type. Once defined, this new type can be used to create objects of that type. Thus, a class is a template for an object, and an object is an instance of a class.

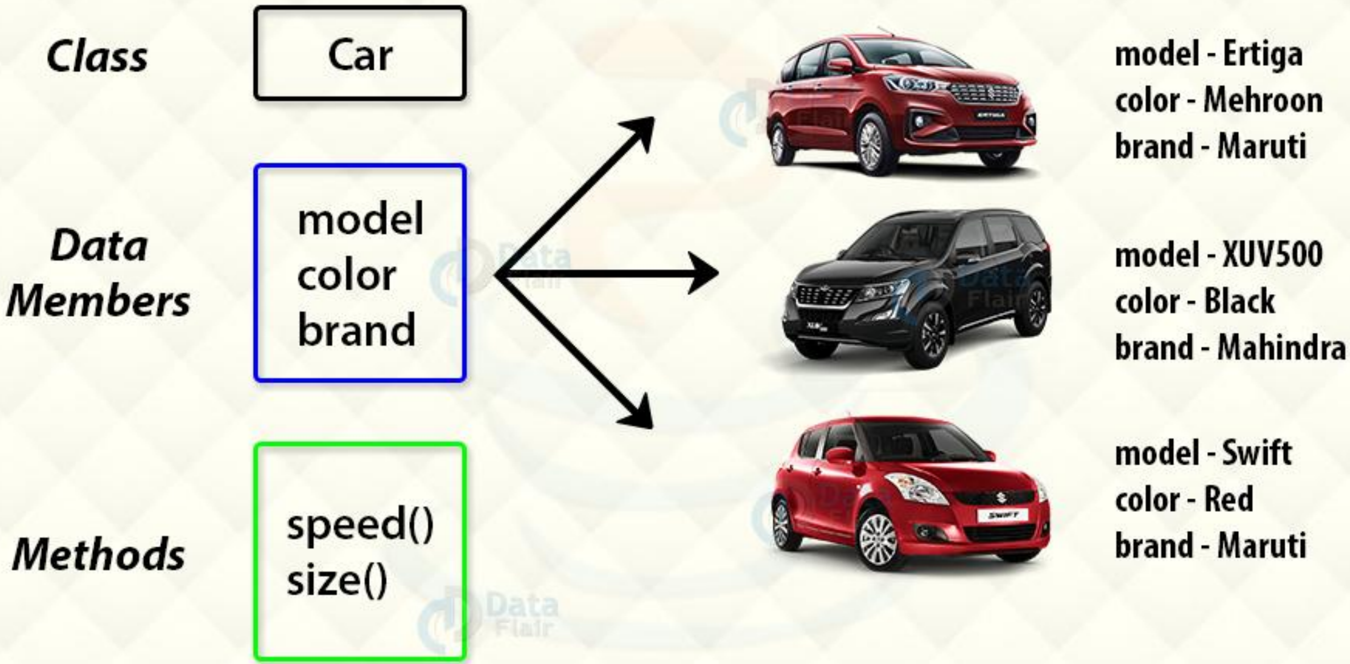
Object

Object is a basic unit of Object-Oriented Programming and represents the real life entities.

An object consists of :

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

Classes & Objects in Java



Class

Car

Data Members

model
color
brand

Methods

speed()
size()

model - Ertiga
color - Mehroon
brand - Maruti

model - XUV500
color - Black
brand - Mahindra

model - Swift
color - Red
brand - Maruti

Installation

https://docs.google.com/document/d/1MrsuUJ05V6GazqPcjNPifAi3UjLHhCnU4QiN-4fvJyc/edit?usp=sharing

First Java program

```
class FirstProgram {

        public static void main(String args[]) {

                System.out.println("Hello MNNIT");

        }

}
```

How to create objects in Java?

Suppose you have a class MyFirst and you want to create its object. You can do this by simply writing:

```
MyFirst obj = new MyFirst();
```

1. MyFirst written first time is the type of reference variable created to store object reference.
2. obj is the name of reference variable.
3. MyFirst written second time is the type of object created.

Home Assignment

What are other ways to create objects in Java?

Second Java Program

2021-22-Classes/SecondProgram.java at main · CC-MNNIT/2021-22-Classes (github.com)

Constructor

A constructor initializes an object immediately upon creation. It has the same name as the class in which it resides and is syntactically similar to a method.

The constructor is automatically called when the object is created, before the new operator completes.

They have no return type, not even void.

Third Program

2021-22-Classes/ThirdProgram.java at main · CC-MNNIT/2021-22-Classes (github.com)

‘this’ keyword

Sometimes a method will need to refer to the object that invoked it. To allow this, Java defines the this keyword. this can be used inside any method to refer to the current object.

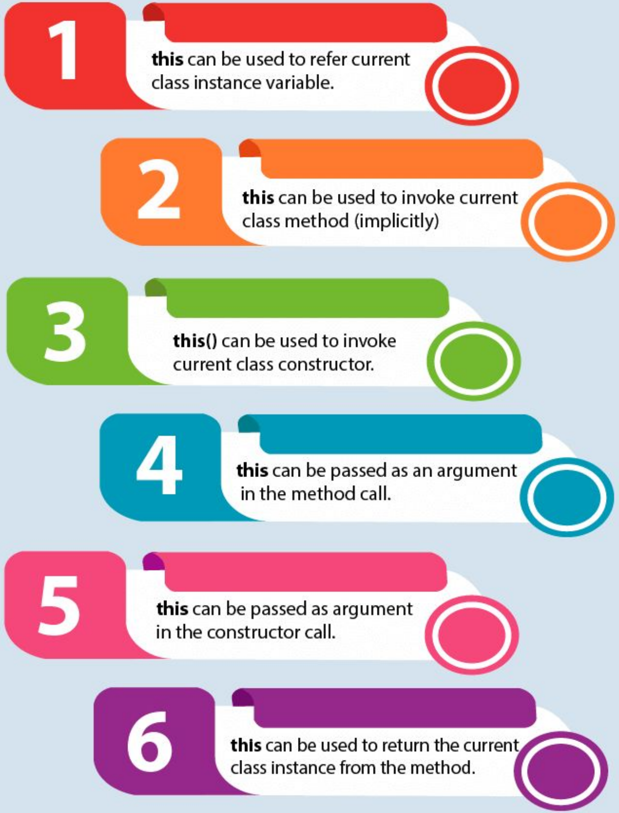
This is always a reference to the object on which the method was invoked. You can use this anywhere a reference to an object of the current class’ type is permitted.

Improvement in third Program

```
Box(double width, double height, double depth) {
    this.width = width;
    this.height = height;
    this.depth = depth;
}
```

Usage of java this keyword

There can be a lot of usage of java this keyword. In java, this is a reference variable that refers to the current object.



1. **this** can be used to refer current class instance variable.
2. **this** can be used to invoke current class method (implicitly)
3. **this()** can be used to invoke current class constructor.
4. **this** can be passed as an argument in the method call.
5. **this** can be passed as argument in the constructor call.
6. **this** can be used to return the current class instance from the method.

Java's syntax is similar to C: for loop

```
class Demo {
    public static void main(String args[]) {
        for(int i = 1; i <= 10; ++i) {
            System.out.println(i);
        }
    }
}
```

Java’s syntax is similar to C: while loop

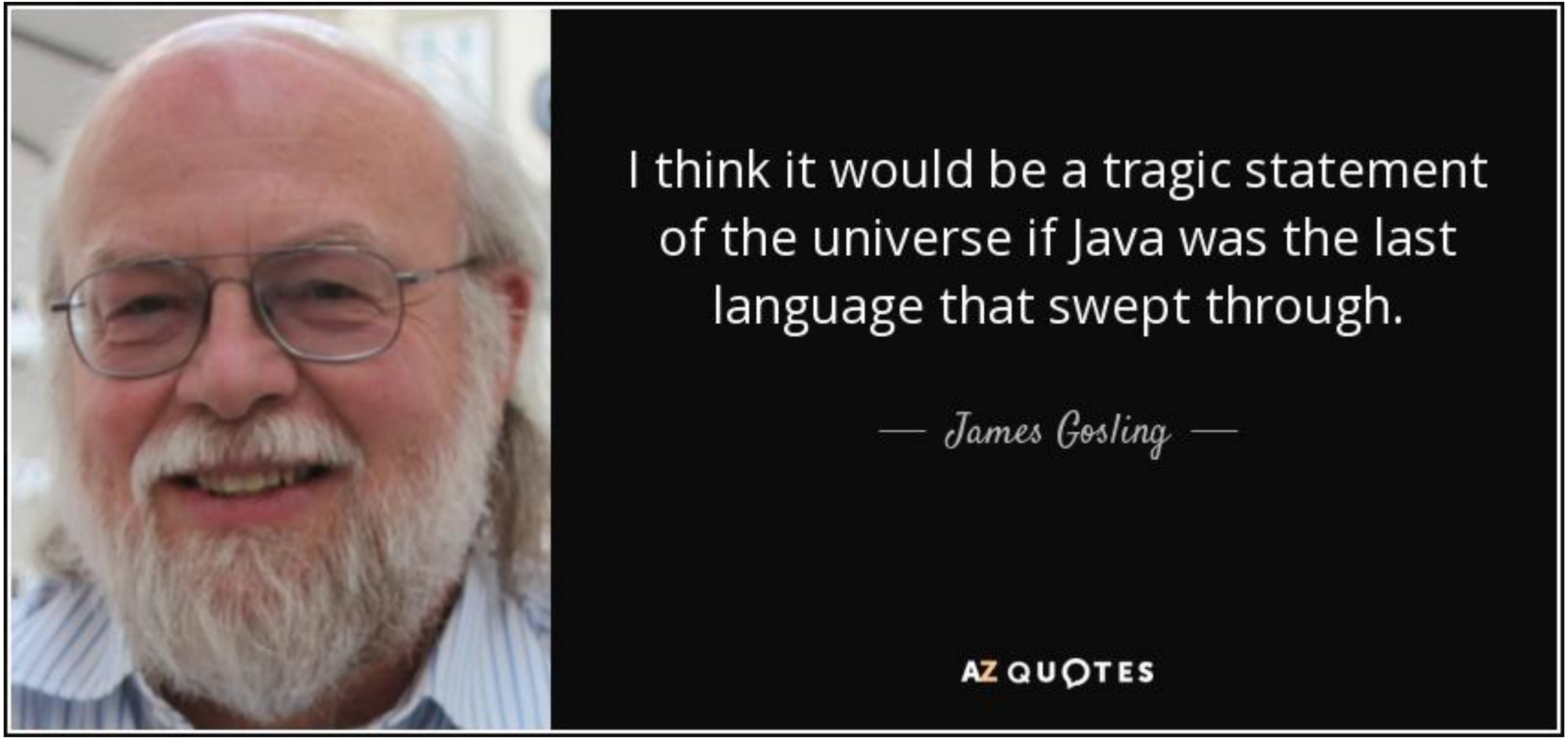
```
class Demo {
    public static void main(String args[]) {
        int n = 10;
        while(n-- > 0) {
            System.out.println(n);
        }
    }
}
```

Java’s syntax is similar to C: do… while loop

```
class Demo {
    public static void main(String args[]) {
        int n = 10;
        do {
            System.out.println(n);
        } while(n-- > 0);
    }
}
```

Java’s syntax is similar to C: if… else if… else

```
class Demo {
        public static void main(String args[]) {
                int n = 10;
                if(n % 3 == 0)
                        System.out.println(“Divisible by 3”);
                else if(n % 2 == 0)
                        System.out.println(“Divisible by 2”);
                else
                        System.out.println(“Neither divisible by 2 nor by 3”);
        }
}
```



I think it would be a tragic statement of the universe if Java was the last language that swept through.

— *James Gosling* —

AZ QUOTES

Thank You!