TODAY'S AGENDA

- Naming Conventions
- Overview of Comment, Package, import, Class, Main class (main method) and Access Modifiers.
- First Program
- How Java works (Architecture)?
- What is Variable? (declaration and initialisation)
- What is Constants?
- Constants vs Variables
- Difference between local and global variables
- Identifiers in Java

Naming Conventions

- All Java **class** names, **interfaces**, **abstract** classes names and **enum** must be started with Upper case letters and subsequent symbols must also be Upper case letters.
- All Java **variables** names must be started with lower case letters and the subsequent symbols must be upper case letters.
- All Java **method** names must be started with lower case letters but the subsequent symbols must be uppercase letters.
- All Java **constants** must be provided in Upper case letters.
- All Java **package** names must be provided in lower case letters.

Comment

- Single Line Comment
- Multi-line Comment
- Documentation Comment

```
1  // a single line comment is declared like this
2  /* a multi-line comment is declared like this
3  and can have multiple lines as a comment */
4  /** a documentation comment starts with a delimiter and ends with */
```

Package

Package Section:

- package is the collection of related classes and interfaces as a single unit.
- Predefined package (by Java PL and provided along with Java software). java.io, java.util, java.sql
- User defined package (by developer as per application requirement).
- package is a keyword and any keyword in java is written in lowercase letters.

Import Statement

- To make available classes and interfaces of other packages into the present Java file, so it can be used.
- We can import a specific class or classes in an import statement.
- An import statement is always written after the package statement but it has to be before any class declaration.

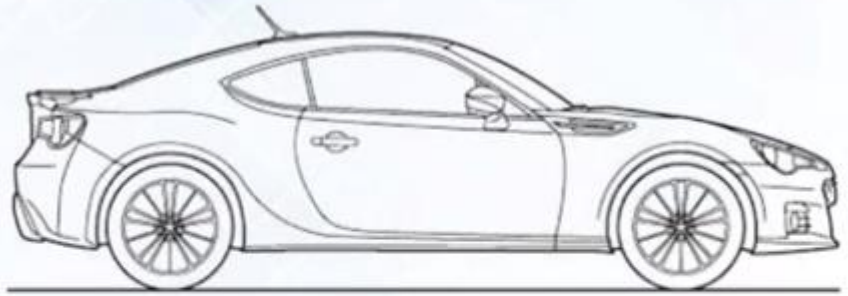
```
1  import java.util.Date; //imports the date class
2  import java.applet.*;  //imports all the classes from the java applet package
```

Class

Classes

- A class in java is a blueprint which includes all the data.
- It describes the state and behavior of a specific object.

Example :



Syntax :

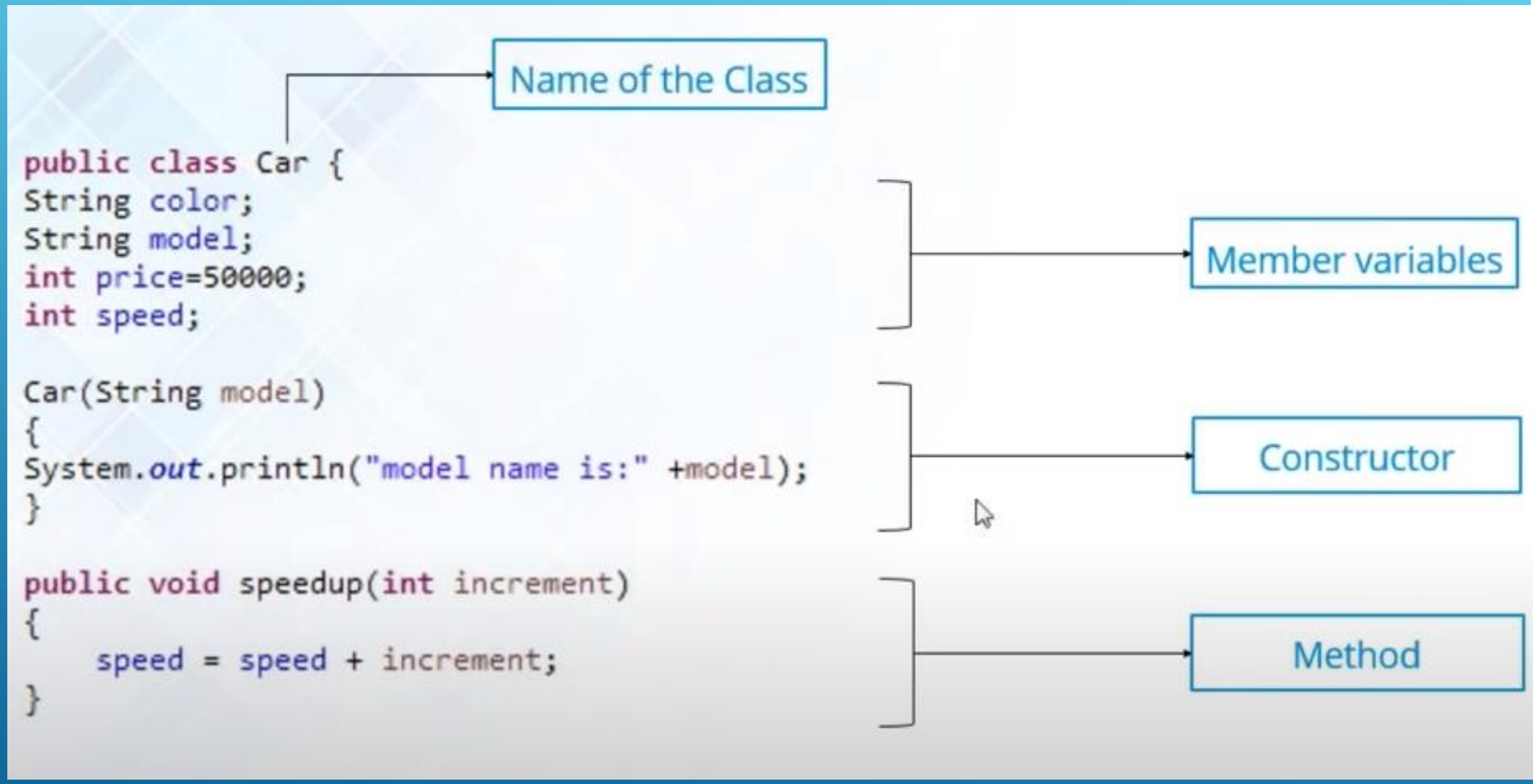
```
Public class Car {

Color
Model
Price
speedUp();
gearChange();
}
```

Class name

Variables

Methods

Structure of a Class



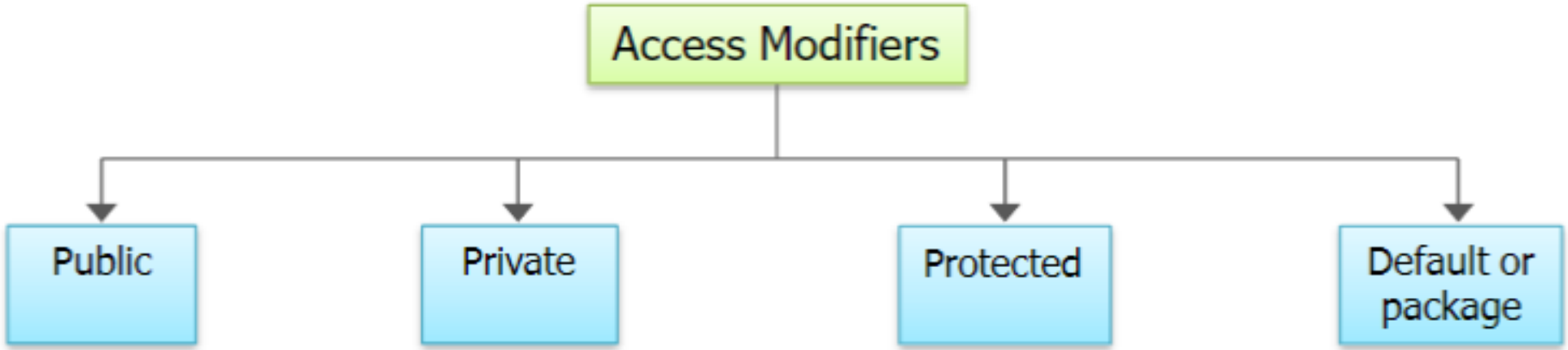
Main Class

Main Class Section:

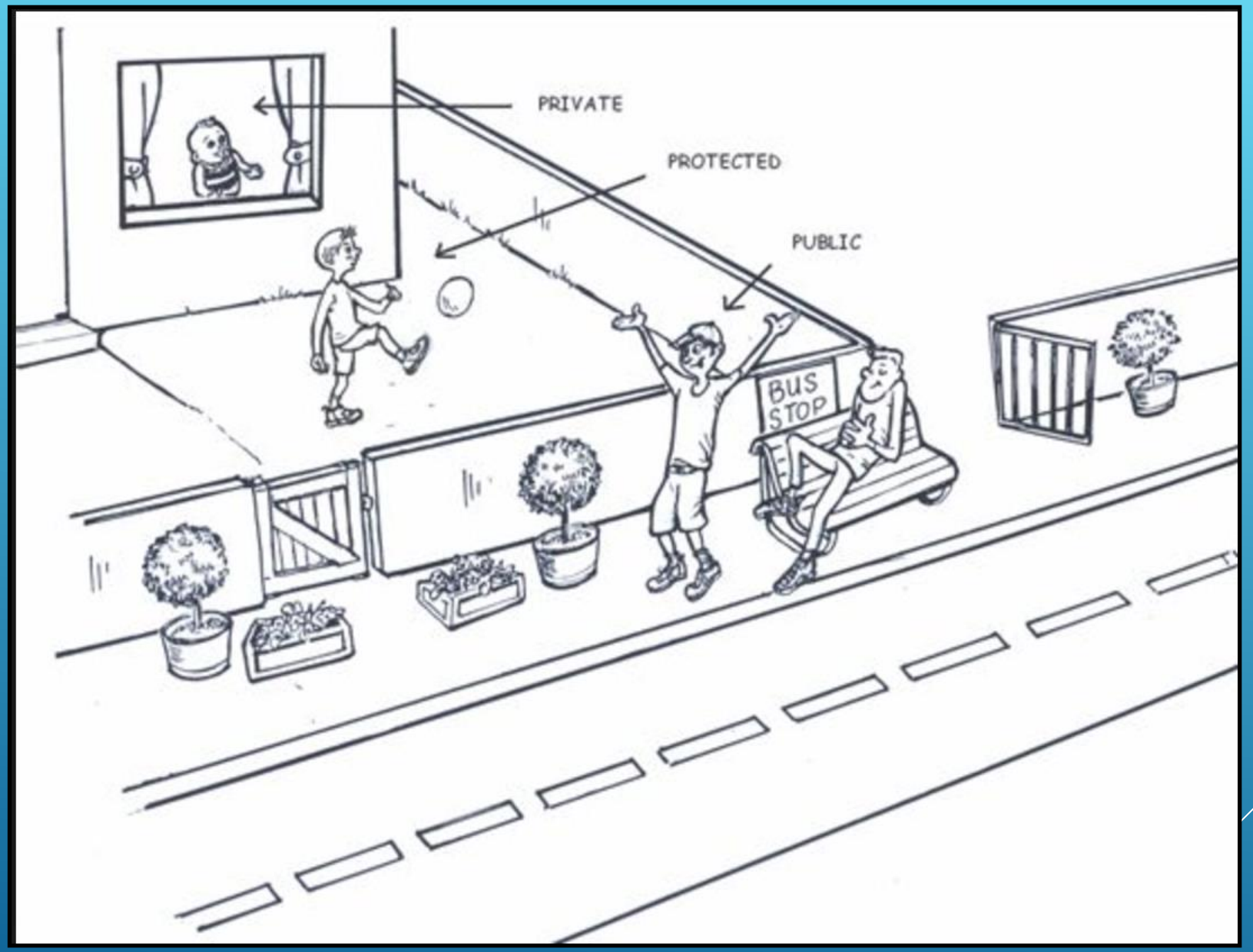
Main Class is a java class, it includes main() method.

The main intention of **main() method** is:

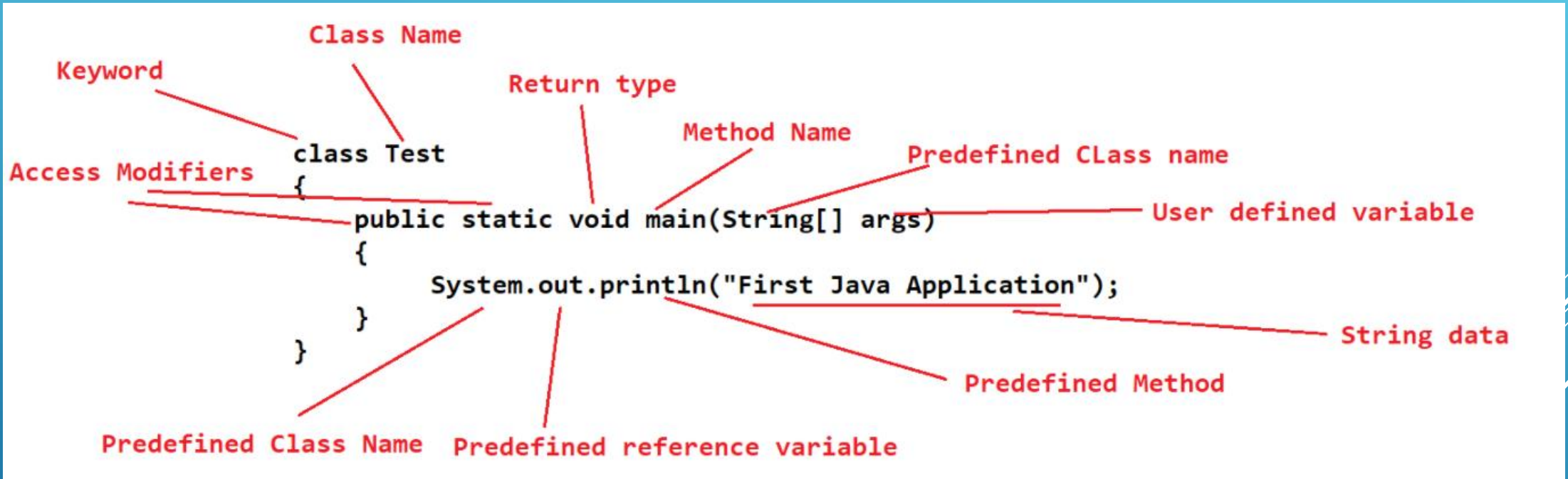
- To manage application logic which we want to execute by JVM directly we have to use main() method.
- To define starting point and ending point to the application execution we have to use main() method.



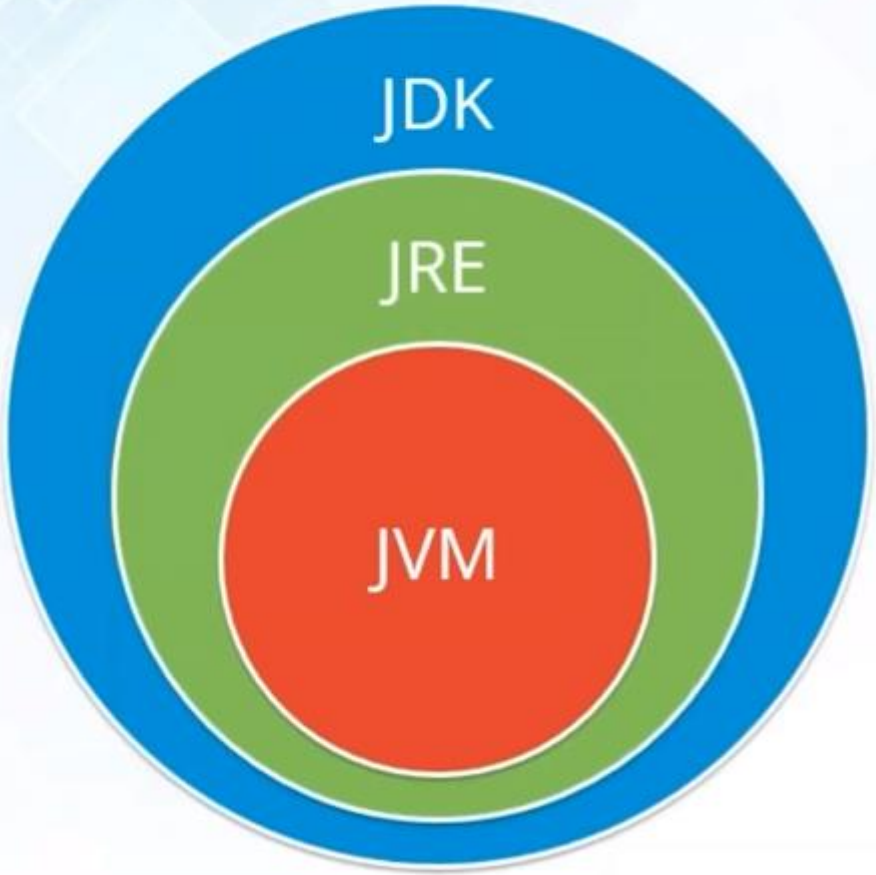
Access Modifiers	Same Class	Same Package	Sub Class	Other Packages
Public	Y	Y	Y	Y
Private	Y	N	N	N
Protected	Y	Y	Y	N
Default	Y	Y	N	N


PRIVATE
PROTECTED
PUBLIC
BUS STOP

My First Program



Java Architecture

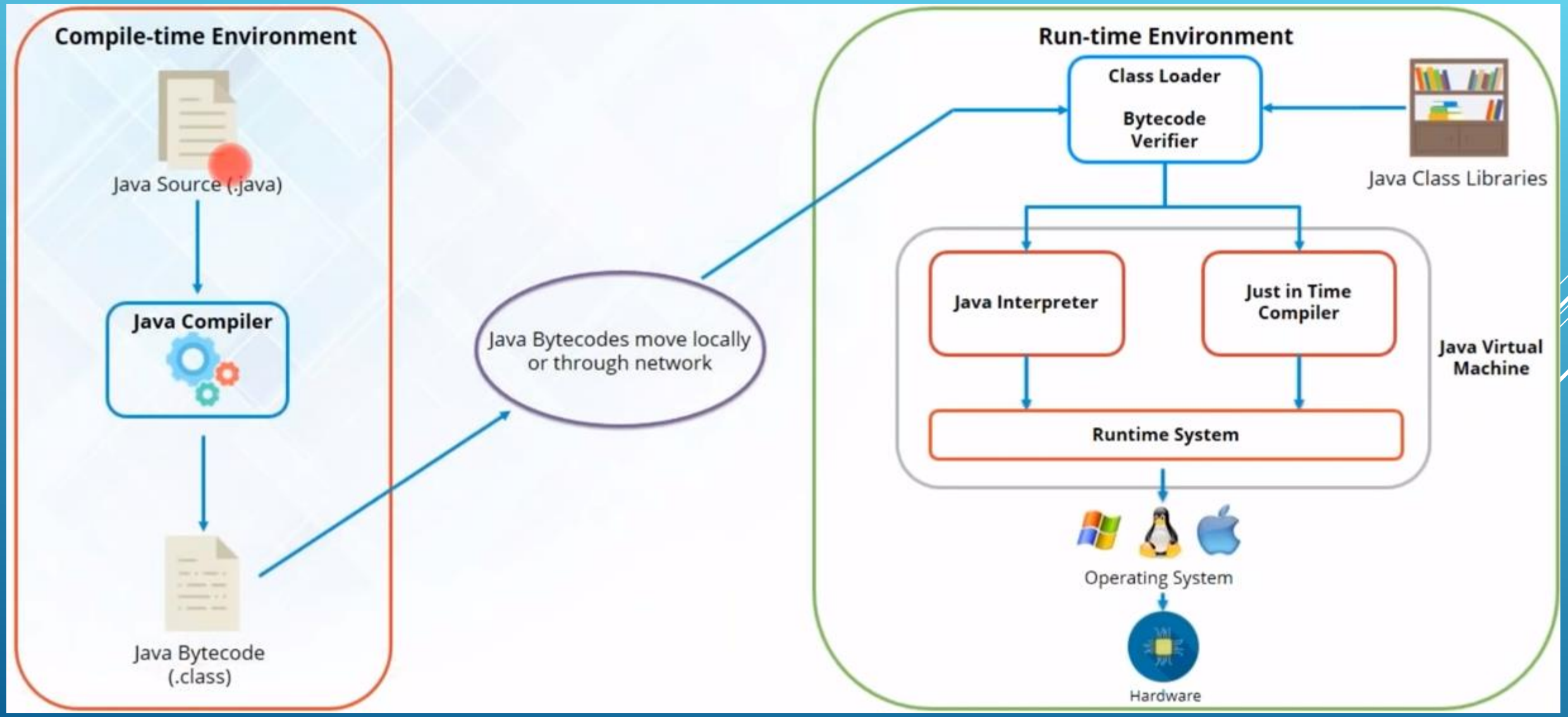


JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

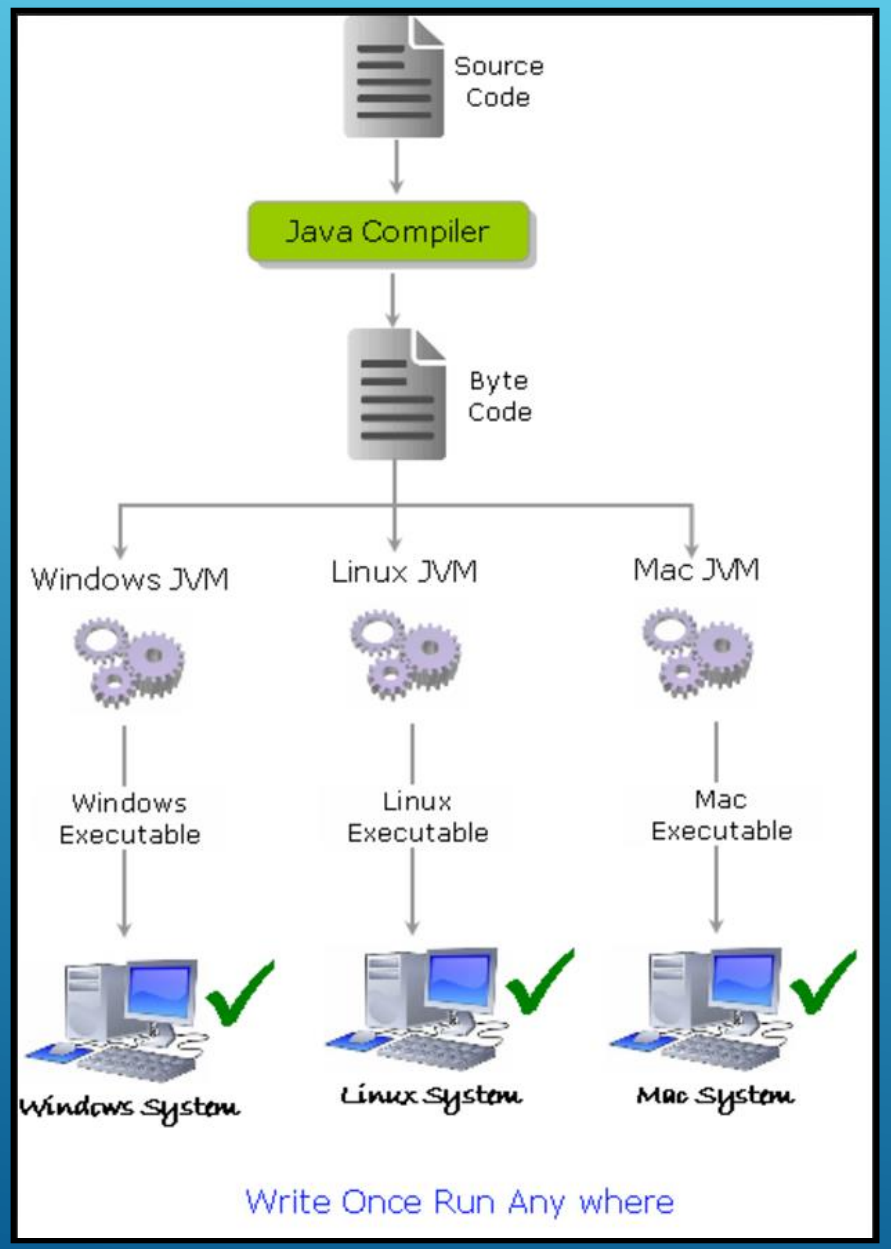
JRE (Java Runtime Environment) is a runtime environment which implements JVM and provides all class libraries and other files that JVM uses at runtime.

JDK(Java Development Kit) is the tool necessary to compile, document and package Java programs. The JDK completely includes JRE.

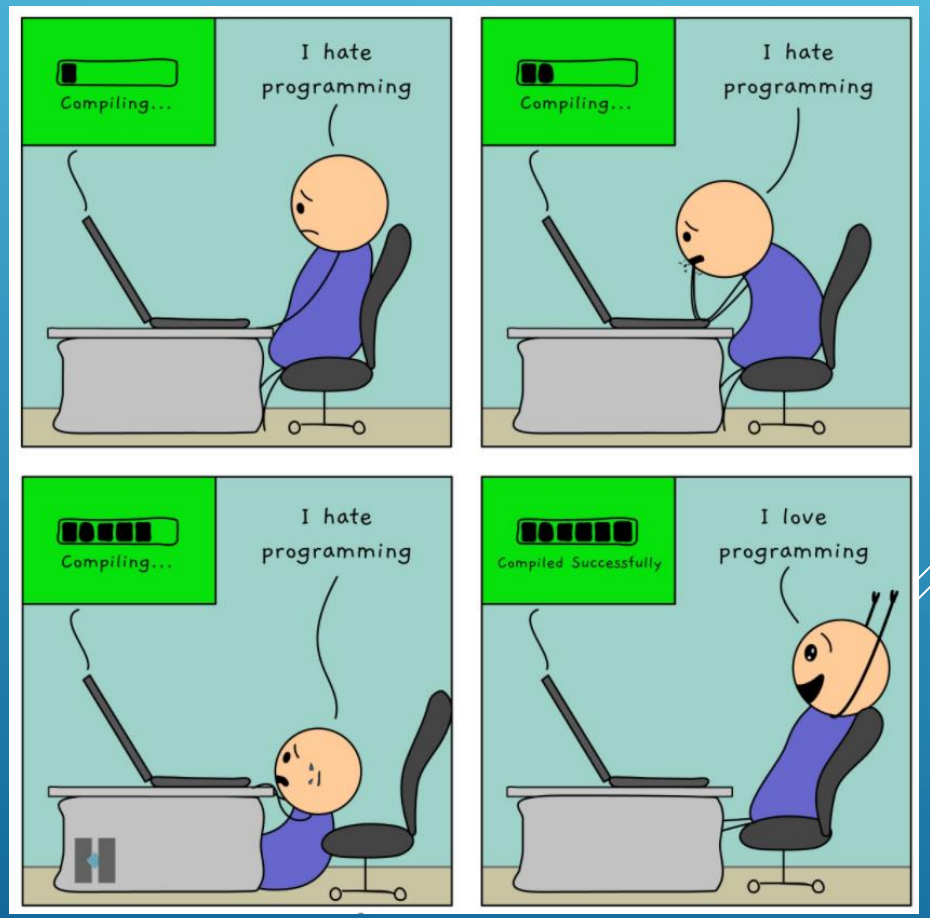
Java Architecture



Java Architecture

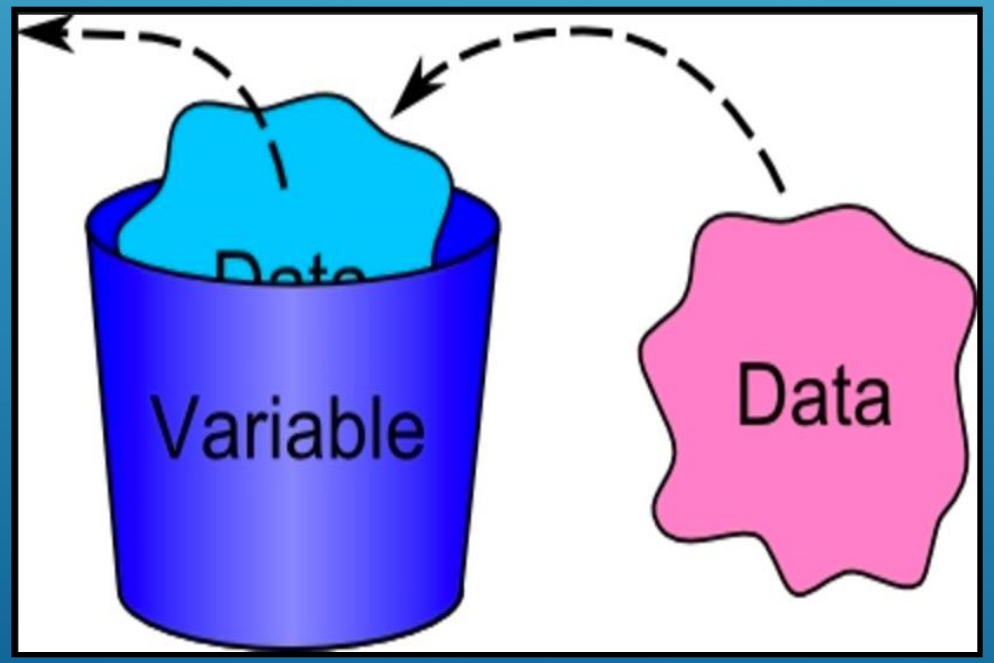


Bytecode is an intermediate code which gets generated when a Java file is compiled using a Javac compiler. After compilation .class file is generated which contains the byte code. This code is platform independent.



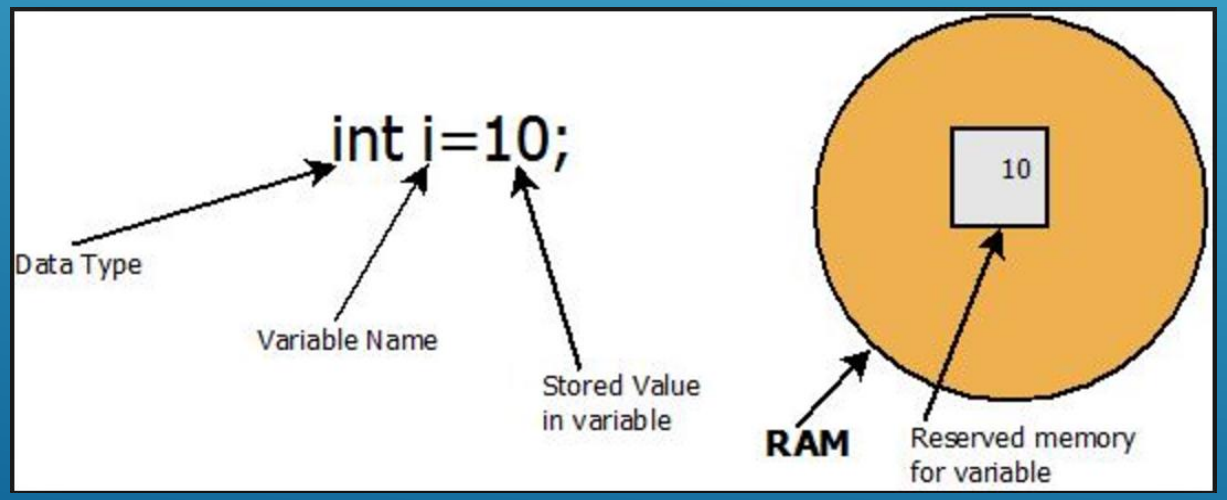
Java Variable

- Variables are nothing but reserved memory locations to store data values, by using this values we are achieving the project requirements. This can be any kind of information ranging from texts, numbers, sentences, etc.
- This means that when you create a variable you reserve some space in the memory. A variable thus has a data type.
- In simple words we can say variables are containers that holds the data.



Variable Declaration and Initialisation

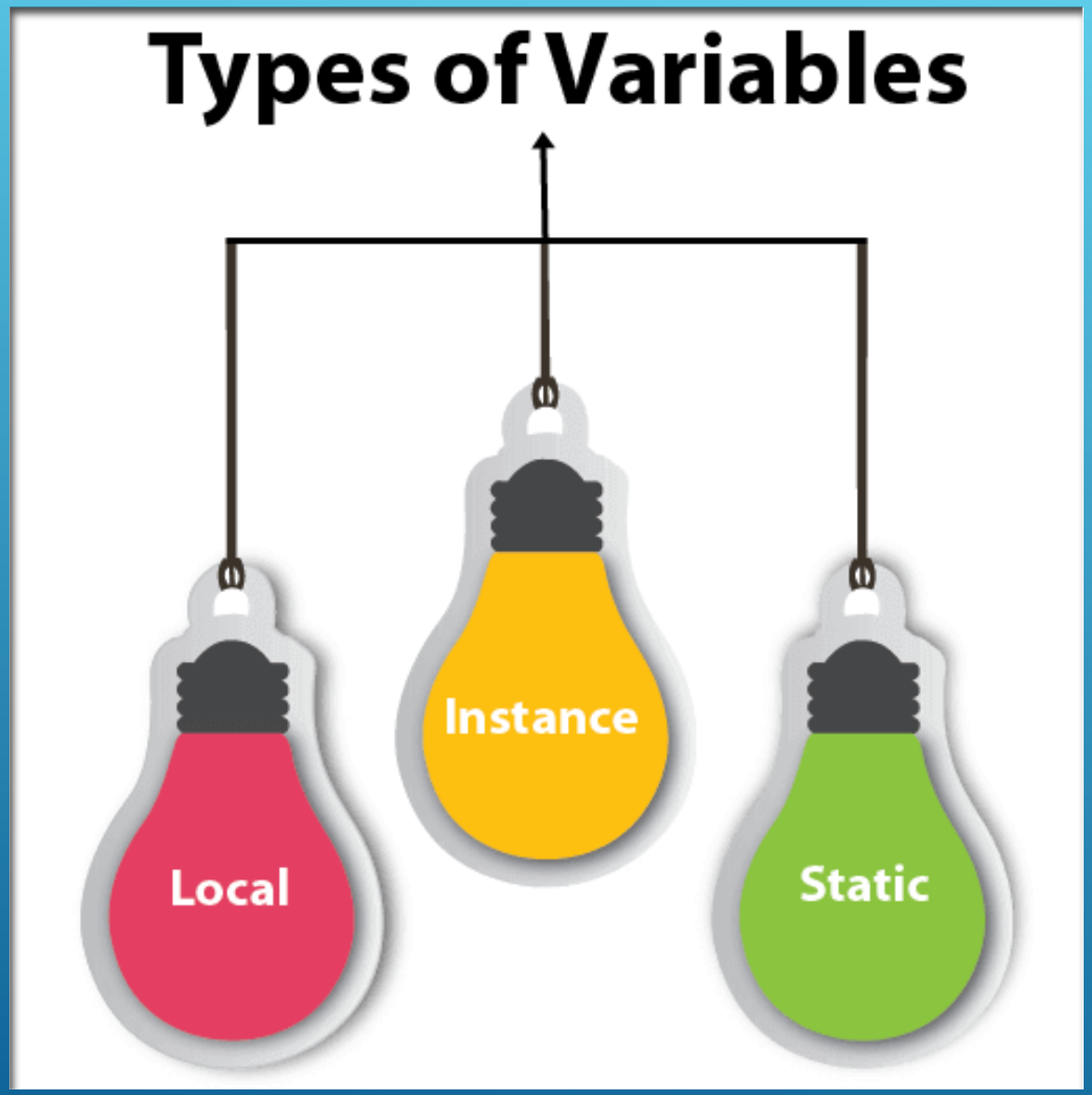
- A variable must be declared by specifying the variable's name and the type of information that it will hold.
- When we assign a value to variable called - variable initialisation
- Variables are containers for storing data values
- When a variable is referenced in a program (by variable name), its current value is used
- Syntax : type variable = value;



data type variable name

```
int total;
int count, temp, result;
```

Multiple variables can be created in one declaration



Local Variable: A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

Instance Variable: A variable declared inside the class but outside the body of the method, is called instance variable. It is not declared as static.

It is called instance variable because its value is instance specific and is not shared among instances.

Static variable: A variable which is declared as static is called static variable. It cannot be local. You can create a single copy of static variable and share among all the instances of the class. Memory allocation for static variable happens only once when the class is loaded in the memory.

Types of Variables

Local

```
class EmployeeId {
   public void EmployeeId()
   {
      // local variable ID
      int id = 0;
      id = id + 6;
      System.out.println("Employee ID : " +
id);
   }

   public static void main(String args[])
   {
      EmployeeId obj = new EmployeeId();
      obj.EmployeeId();
   }
}
```

Instance

```
class Price {
   // Instance variables that are declared in a class
and not inside any function
   int guitarPrice;
   int pianoPrice;
   int flutePrice;
}
public class Main {
   public static void main(String args[])
   {
      Price ob1 = new Price();
      ob1.guitarPrice = 10000;
      ob1.pianoPrice = 5000;
      ob1.flutePrice = 1000;
      System.out.println("Price for second object:");
      System.out.println(ob1.guitarPrice);
      System.out.println(ob1.pianoPrice);
      System.out.println(ob1.flutePrice);

   }
}
```

Static

```
class Manager {

   // static variable salary
   public static double salary;
   public static String name = "Jonathan";
}
public class Main {
   public static void main(String args[])
   {

      // accessing static variable without
object
      Manager.salary = 90000;
      System.out.println(Manager.name + "'s
avg salary:"
                  + Manager.salary);
   }
}
```

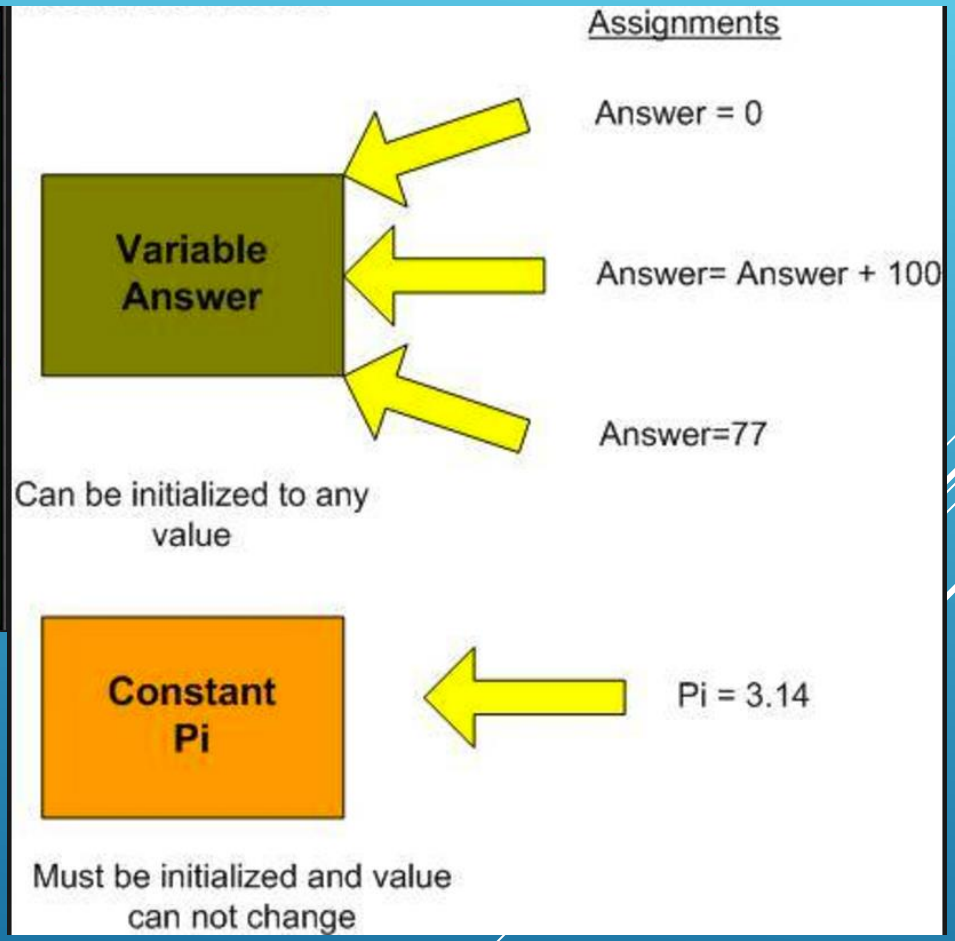
CONSTANTS

- Constants in java refers to the fixed values, which do not change during execution of the program.
- A static final variable is effectively a constant.
- Java constants are normally declared in ALL CAPS. Words in Java constants are normally separated by underscores.
- An example of constant declaration in Java is written below:

```
public class MaxUnits {
   public static final int MAX_UNITS = 25;
   static final double SALES_TAX_RATE = 0.85;
}
```

CONSTANTS VS VARIABLES

	Constants	Variables
Characteristics	Value is not changeable during the course of the program.	Value can be changed anytime during the course of the program.
Usage	Use constant when you want to declare something that won't be changed midway in your program execution.	Use variable to store data that may or will change during the running of the program.



IDENTIFIERS IN JAVA

- An identifier is a name to represent a variable, object or method.
- Identifiers may contain letters (A-Z and a-z), digit (0-9), underscores (_) and dollar signs($).
- Identifiers cannot start with digit.
- Identifiers should be meaningful. (age, sum, totalVolume, bookPrice).
- Example :
  - Reserved words cannot be used.
  - They cannot start with a digit but digits can be used after the first character (e.g., name1, n2ame are valid).
  - They can start with a letter, an underscore (i.e., "_") or a dollar sign (i.e., "$").
  - You cannot use other symbols or spaces (e.g., "%","^","&","#").

RESERVED WORDS (KEYWORDS)

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
extends	final	finally	float
for	goto	if	implements
import	instanceof	int	interface
long	native	new	package
private	protected	public	return
short	static	strictfp	super
switch	synchronized	this	throw
throws	transient	try	void
volatile	while		

- In the Java programming language, a keyword is one of 50 reserved words that have a predefined meaning in the language; because of this, programmers cannot use keywords as names for variables, methods, classes, or as any other identifier.

HOMEWORK

- Write a Java program to print your ‘First Name' on screen and then print again your ‘Last Name’ on the same line.

- Write a Java program to print a list of manual testing topics you have studied on separate lines.

Deadline: Wednesday Midnight Latest