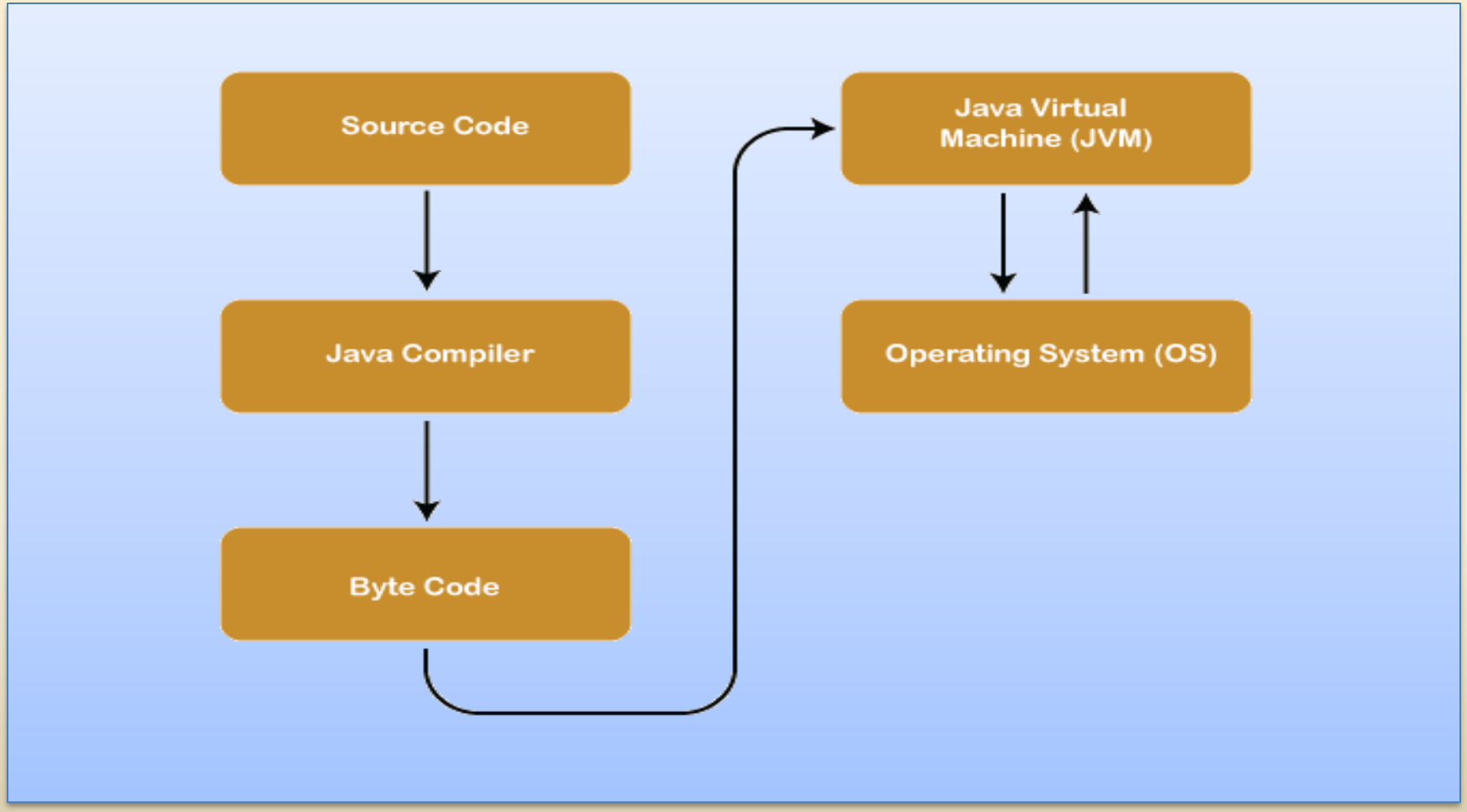
Introduction to Java

- Java is a programming language and computing platform first released by Sun Microsystems in 1995.
- It was created at **Sun Microsystems, Inc.**, where James Gosling led a team of researchers in an effort to create a new language that would allow consumer electronic devices to communicate with each other.

Introduction to Java



Java Virtual Machine

- JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.
- JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).

1. **It is a specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.

2. Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

Features of Java

- **Object Oriented**
- **Platform Independent**
- **Simple**
- **Secure**
- **Portable**
- **Robust**
- **MultiThreaded**
- **Automatic Garbage Collection and Memory Management**

Working with Java

Getting Started

Working with Java

- **Most of the modern applications today are written using Object Oriented Programming**
- **Java has to be written following object oriented programming methodology**

Java Language Features

- **Java strictly follows Object Oriented Programming and supports all features of Object Oriented Programming**
- **It is strongly Typed Language**
- **It simplifies some of the complexities of C++**

Java Language Features

- **Java has support for default and static methods,functional interfaces and lambda expressions,Stream API ,Time API ,Collection,Concurrency and Multithreading**
- **Provides support for Exception Handling**
- **Application for various platforms like window, web ,cloud and mobile can be created**
- **Support for connecting and managing database using JDBC**

Creating classes

- **Let us have a look at how to create a class in Java**

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

Variables and Data Types

- **To store values in programs written in Java,variables are required**
- **A variable is named location inside system in which we can store some value**
- **Value to be stored is based on data type declared along with variable**

Variables and DataType

- **Java supports following Data Types**
  - **Primitive Types**
  - **Reference Types**

Variables and Data Type

Primitive Types in Java

- **int**
- **long**
- **short**
- **byte**
- **char**
- **boolean**
- **double**
- **float**

Variable and Data Type

- **Reference Types**
  - **Class Types**
  - **String**
  - **Array Types**

Variables and Data Type

- **A Variable can be declared inside a class or inside a function /method in a class**
- **A variable declared inside class outside function is a class level variable**
- **A variable declared inside a method is a local variable**

Variable and Data Type

- **Syntax to declare a variable**
  Java

  <access modifier> <data type>[variable name]

Variable and Data Type

- **Lets see an example to declare a variable**

Variable and Data Type

```
public class Calculator
{
   private int first_num;
}
```

Main method in a class

- **Main method is the entry point for a program .**
- **This method is executed first in a program**
- **It has to be kept inside a class**
- **It is a static method**
- **A static method is common to all instances in a program ,and there will be only one copy of this method throughout the execution of program**

Main Method

```
//Main method in Java
public class Calculate
{
 public static void main(String [] args)
 {
 }
 }
```

Creating Objects

- **An object is an instance of a class**
- **Purpose of object creation is to provide memory to class and access members of that class**
- **Object get attributes and behaviour from class**

Creating Objects

- The new operator is used to create an object of class and allocate memory to this object
- Eg:-

  Calculate obj=new Calculate()

- new operator provides for dynamic memory allocation
- Constructor of class has to be used along with new operator to create an instance of class

Creating Objects

```
//Object creation in Java
      public class Calculate
{
 private int first_num;
 static void main(String [] args)
{
  Calculate obj=new Calculate();

}
}
```

Accessing Members of a class

- **Once instance of a class is created ,members can be accessed using object**

  eg:-

  obj.first_name

Accessing members of class

```
//Object creation in Java
      public class Calculate
{
 private int first_num;
 static void main(String [] args)
{
  Calculate obj=new Calculate();
  obj.first_name=100;
}
}
```

Lets Summarize

Exercises

- **Write a program to print below given text on console using Java**

##

Welcome to programming in Java

##

Methods in Java

- **Methods are used to write logic and executable statements for a program**
- **They are part of class and included in any object of class**
- **Methods are also known as procedures or functions in other programming languages**

Methods in Java

- **Syntax of method**

- **[access modifier] [return type] [function name](parameters) {**

//operations

}

Passing Parameters to Method

- **Method can accept different type of parameters that can primitive or reference type**
- **While calling the method the arguments passed should match with function signature**

Constructor in Java

- **A constructor in Java is like a special function**
- **It has a name similar to class**
- **It does not have a return type**
- **It is used to initialize members of class**
- **It is used along with new operator to return instance of class**
- **A constructor can be parameterized or non parameterized**
- **We can have more then one constructor inside a class**
- **A constructor can be declared with access specifiers**
- *If a class has no explicit instance constructors, Java provides a parameterless constructor that you can use to instantiate an instance of that class,*

Constructor in Java

```
class Employee
{
int empid;
string employeename;
 Employee()
{
  empid= 1;
 employeename="Peter Jones";
}
   }
```

Let’s see creation and working of a constructor