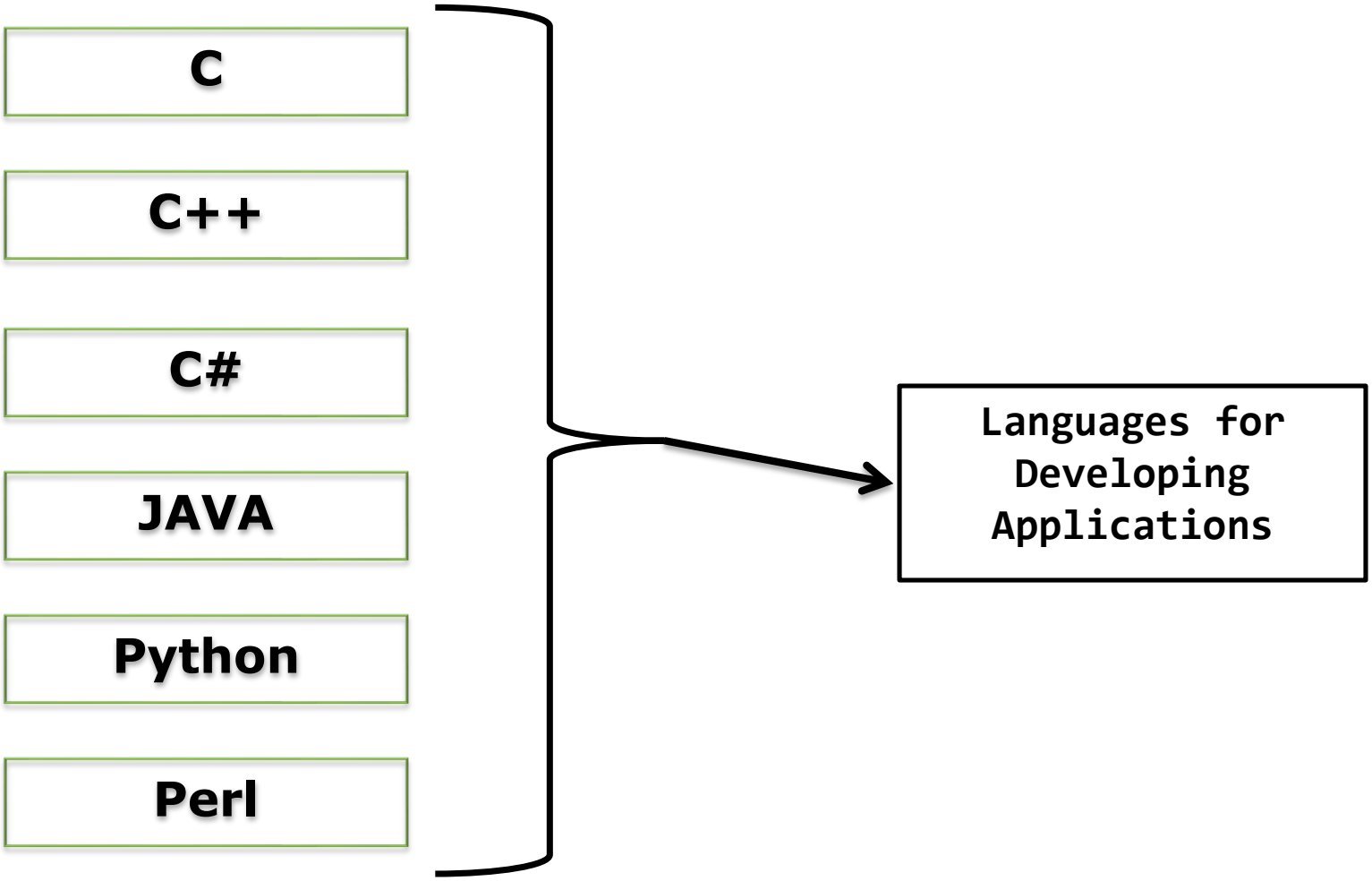
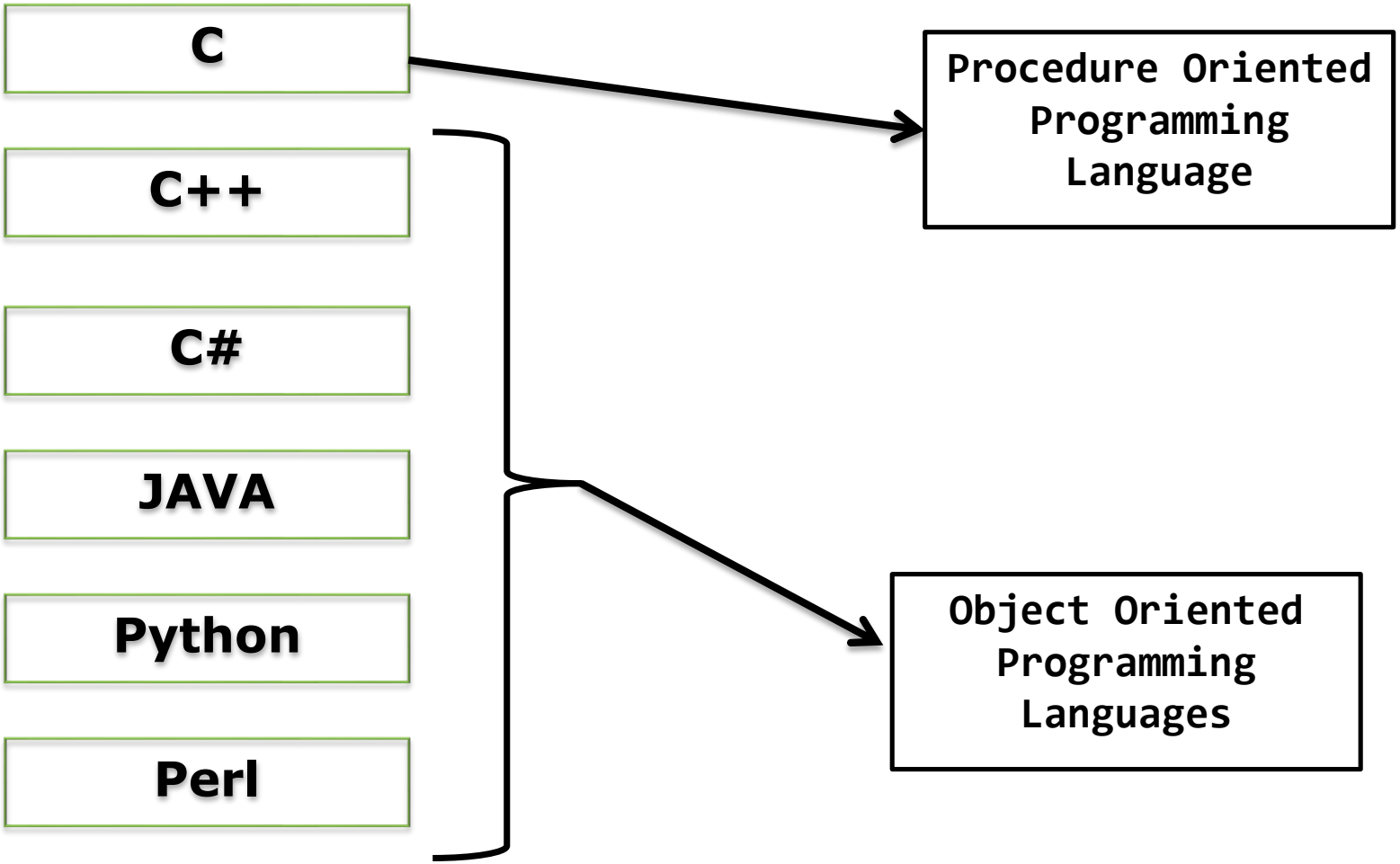
JAVA

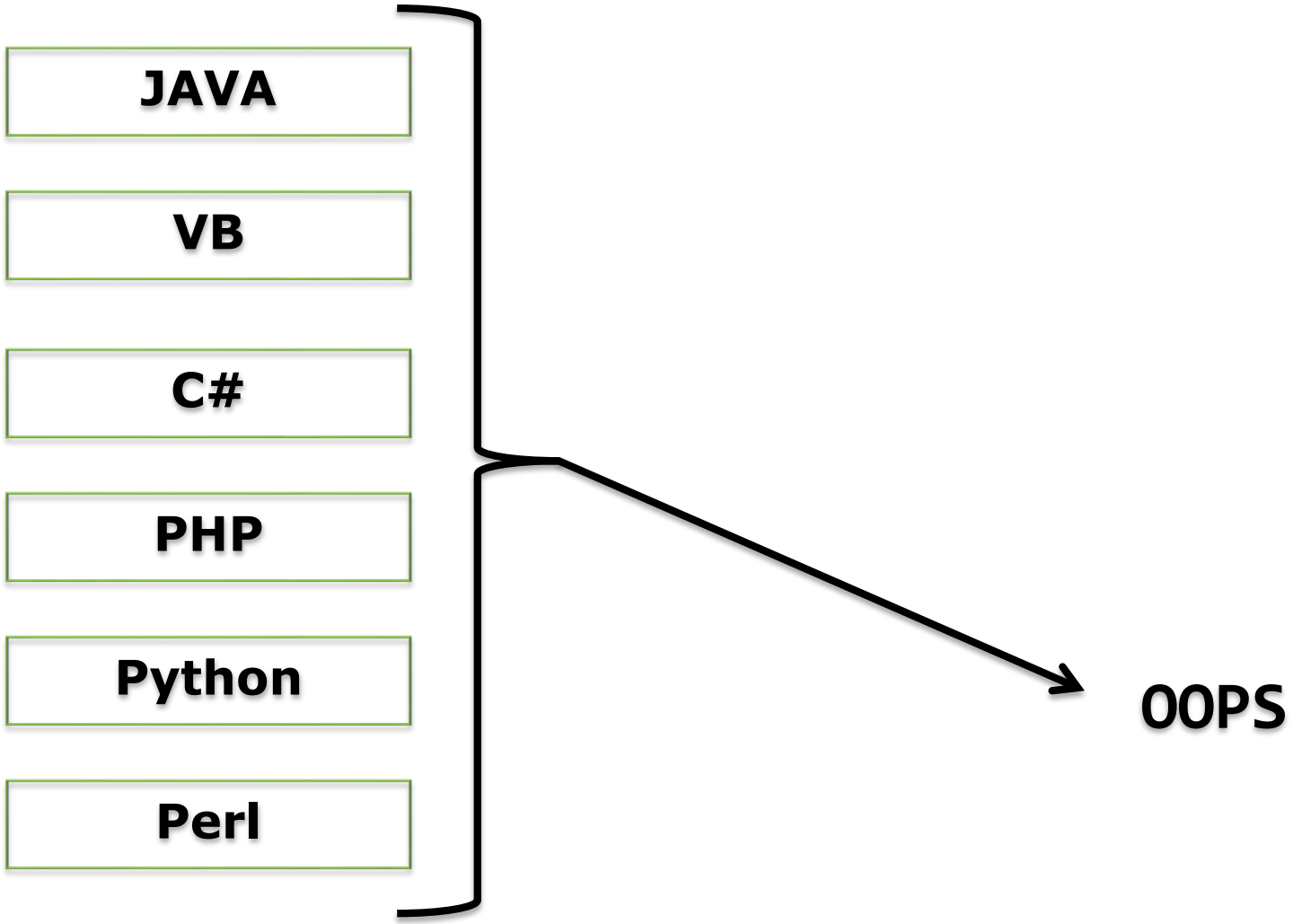
Programming Languages



Programming Languages

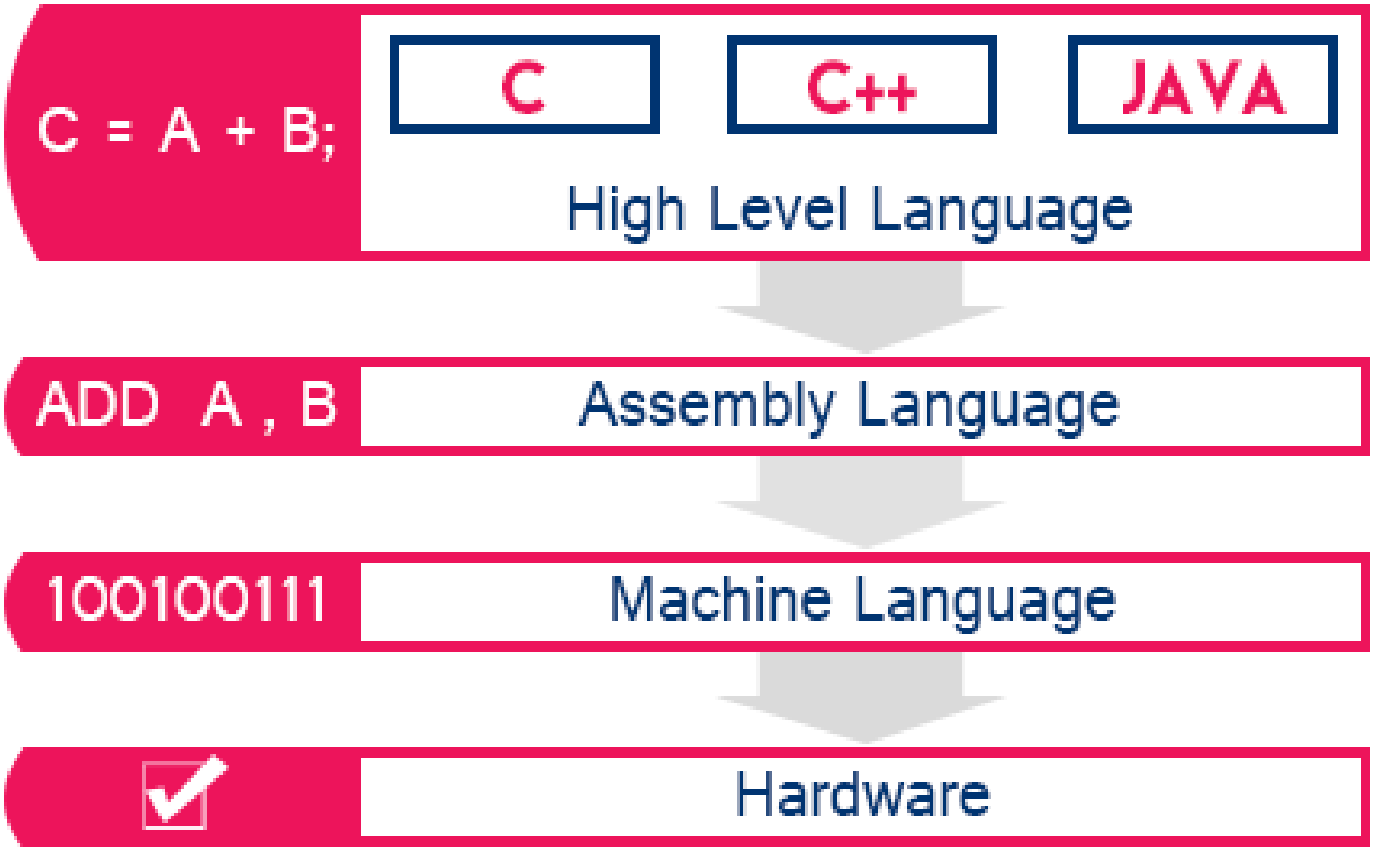


Programming Languages

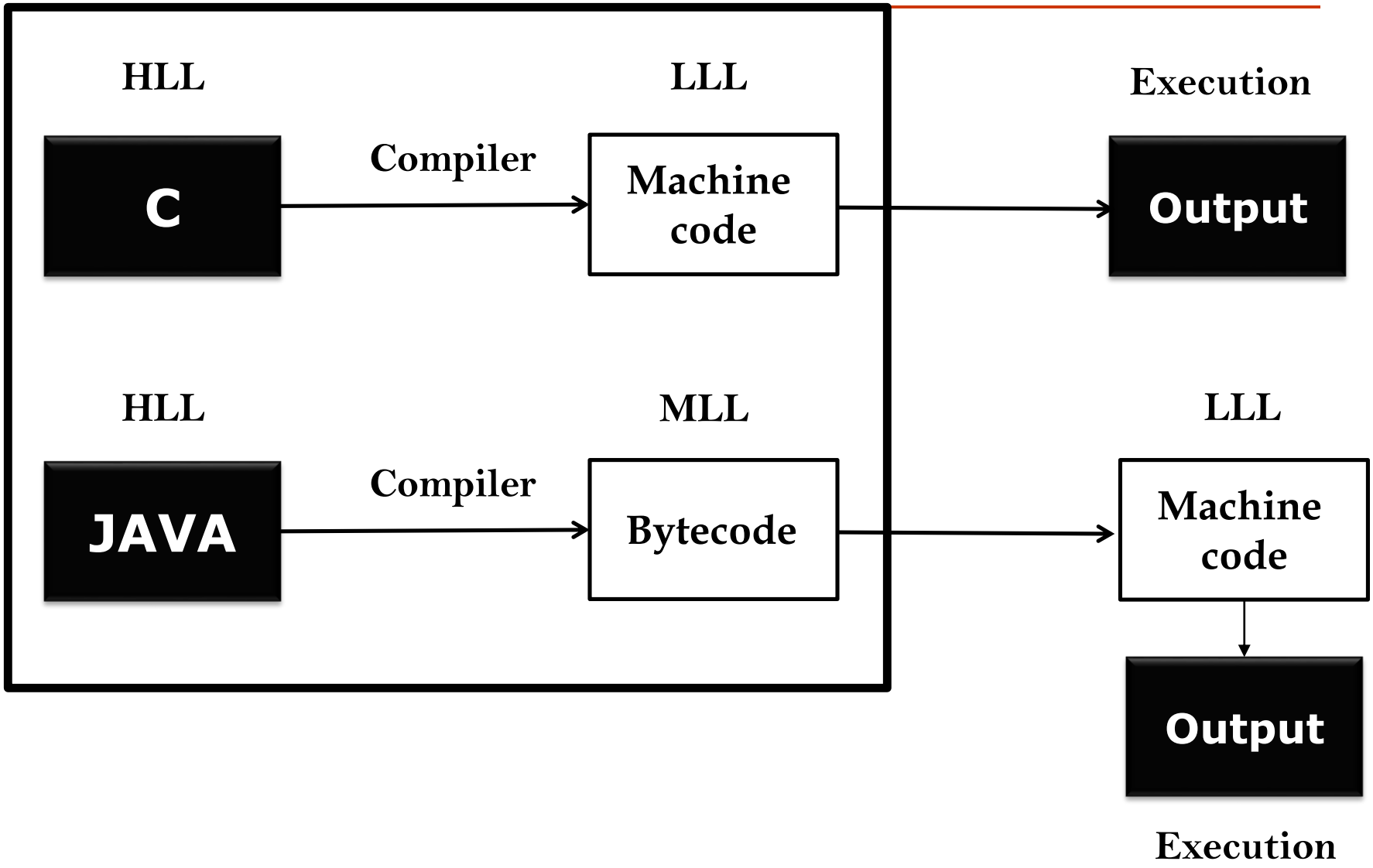


Java Features

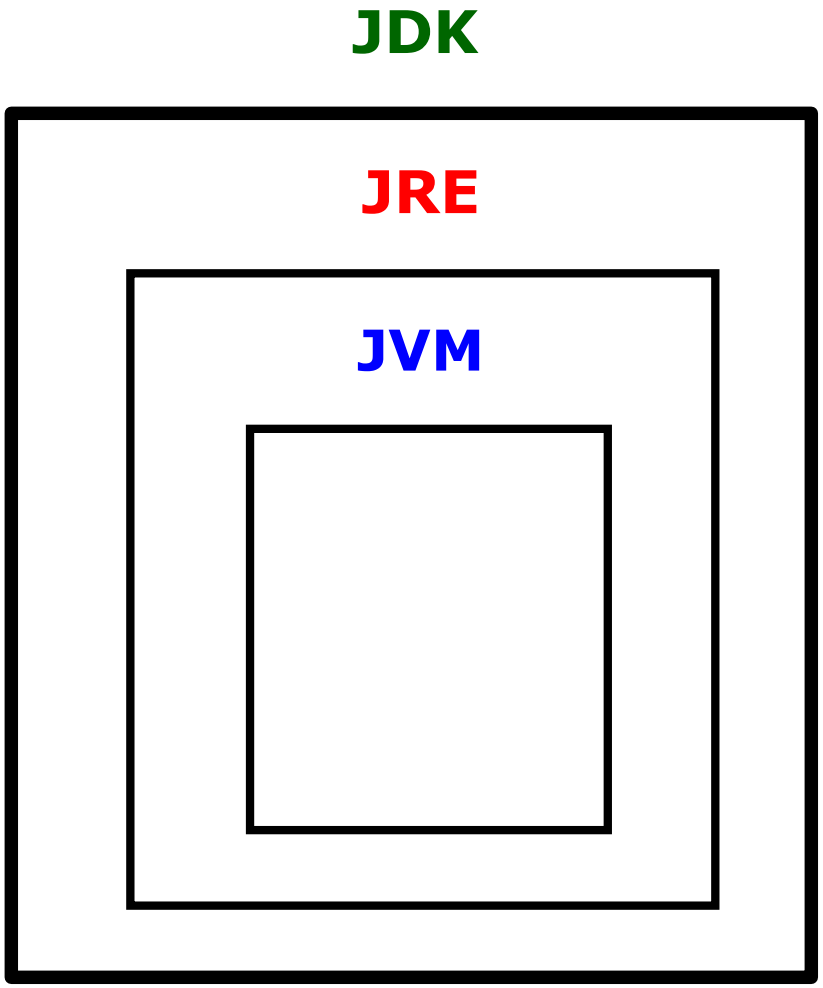
- Platform Independence
- Object Oriented Programming Language


C = A + B;
C
C++
JAVA
High Level Language
ADD A , B
Assembly Language
100100111
Machine Language
Hardware

Programming Execution



JDK Architecture



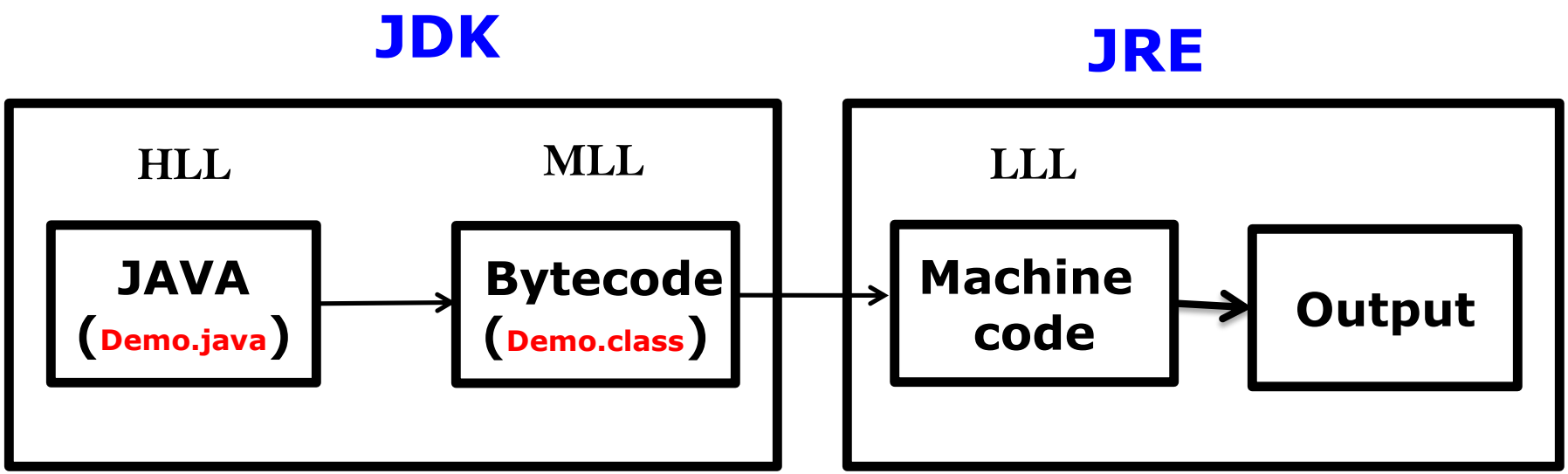
JDK Architecture

JDK → Java Development Kit

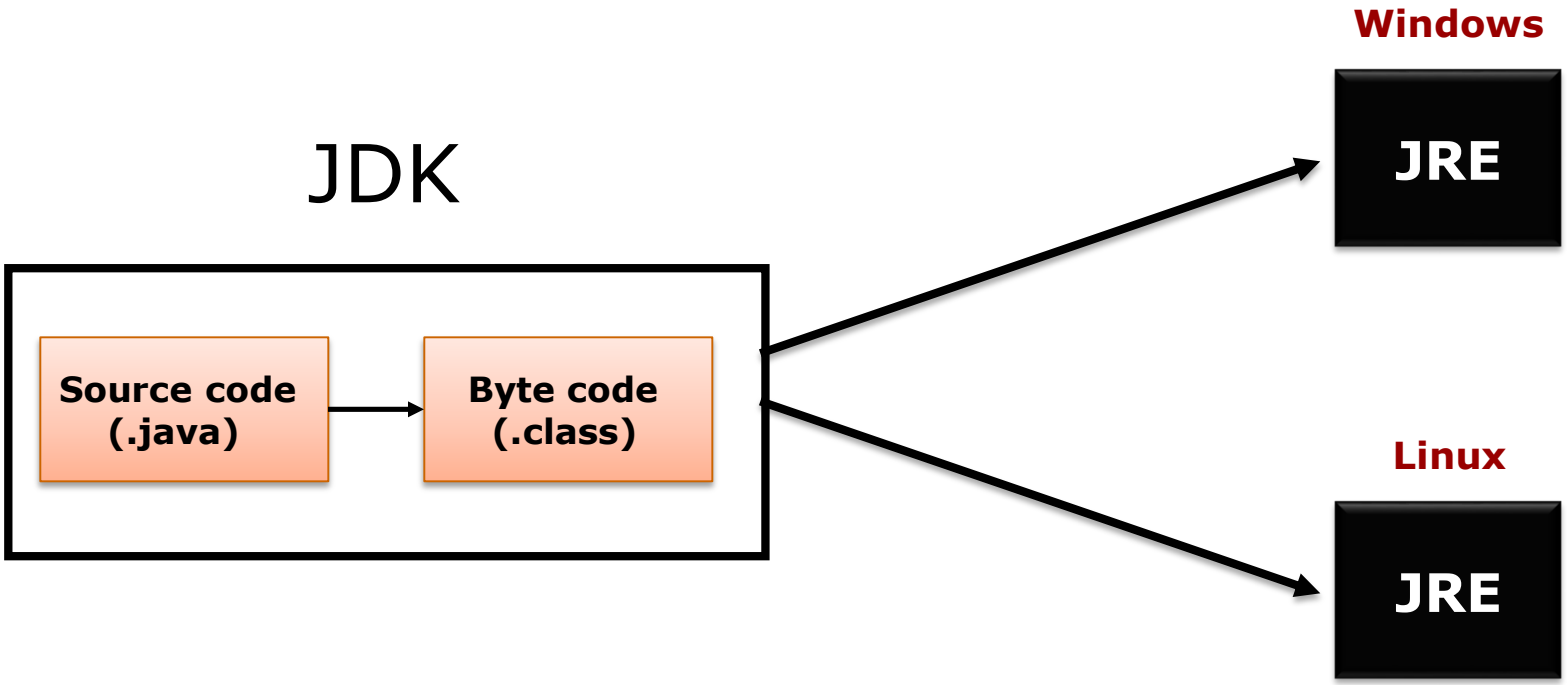
JRE → Java Runtime Environment

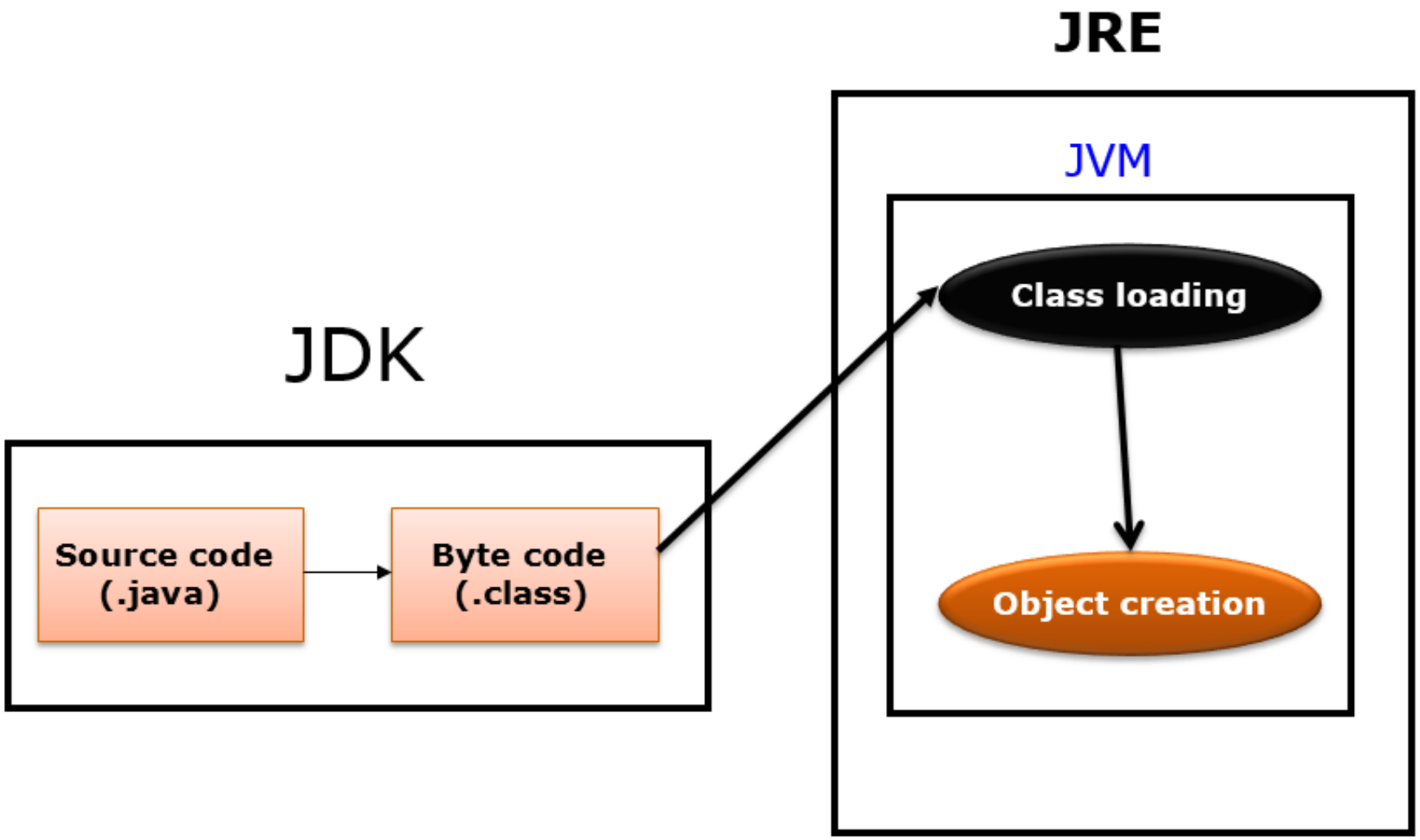
JVM → Java Virtual Machine

Programming Execution



Platform Independence




JRE
JVM
Class loading
Object creation
JDK
Source code
(.java)
Byte code
(.class)

Programming Basics

Printing Statement

```
System.out.println("Welcome");
```

Data Types

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

```
int     a     =  100;
char    b     =  'S';
float   c     =  20.4f;
```

Value Type

```
int     *x =  &a;
char    *y =  &b;
float   *z =  &c;
```

Reference Type

```
int     x    =    100;

char    y    =    'S';
```

Mapping

```
int  id;

void display()
{
  printf(id);
}
```

Data Member

Member Function

```
int  id;

void display()
{
  printf(id);
}
```

Data member → Variable

Function

Member function

Method

Procedure

C Language

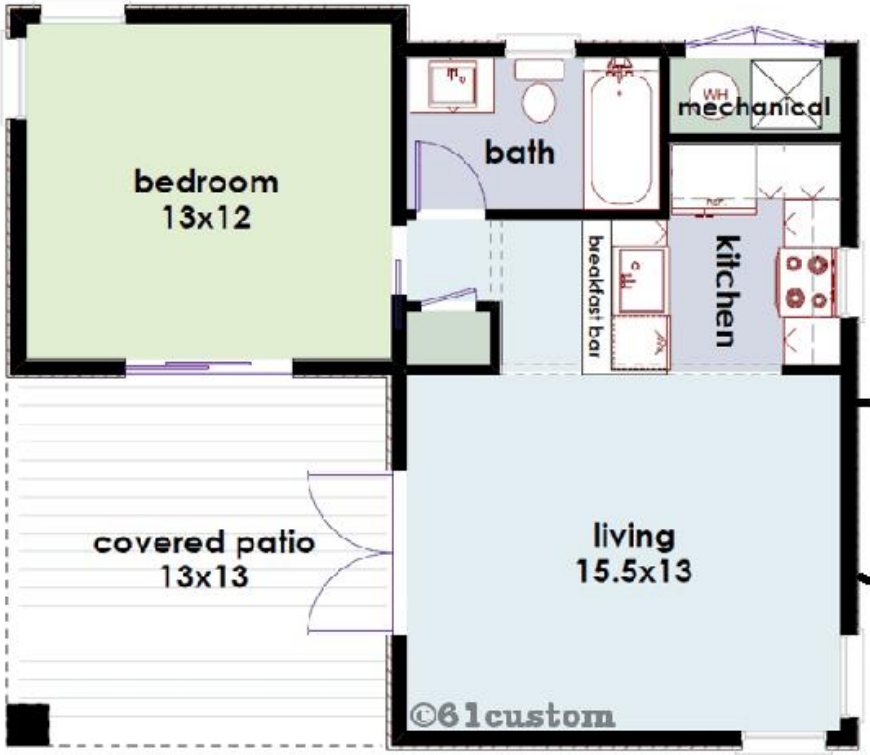
```
void calculator()
{
  // 100 Lines of code
}
void scientific_calculator()
{
  // 200 Lines of code
}
```

Object Oriented Programming Language

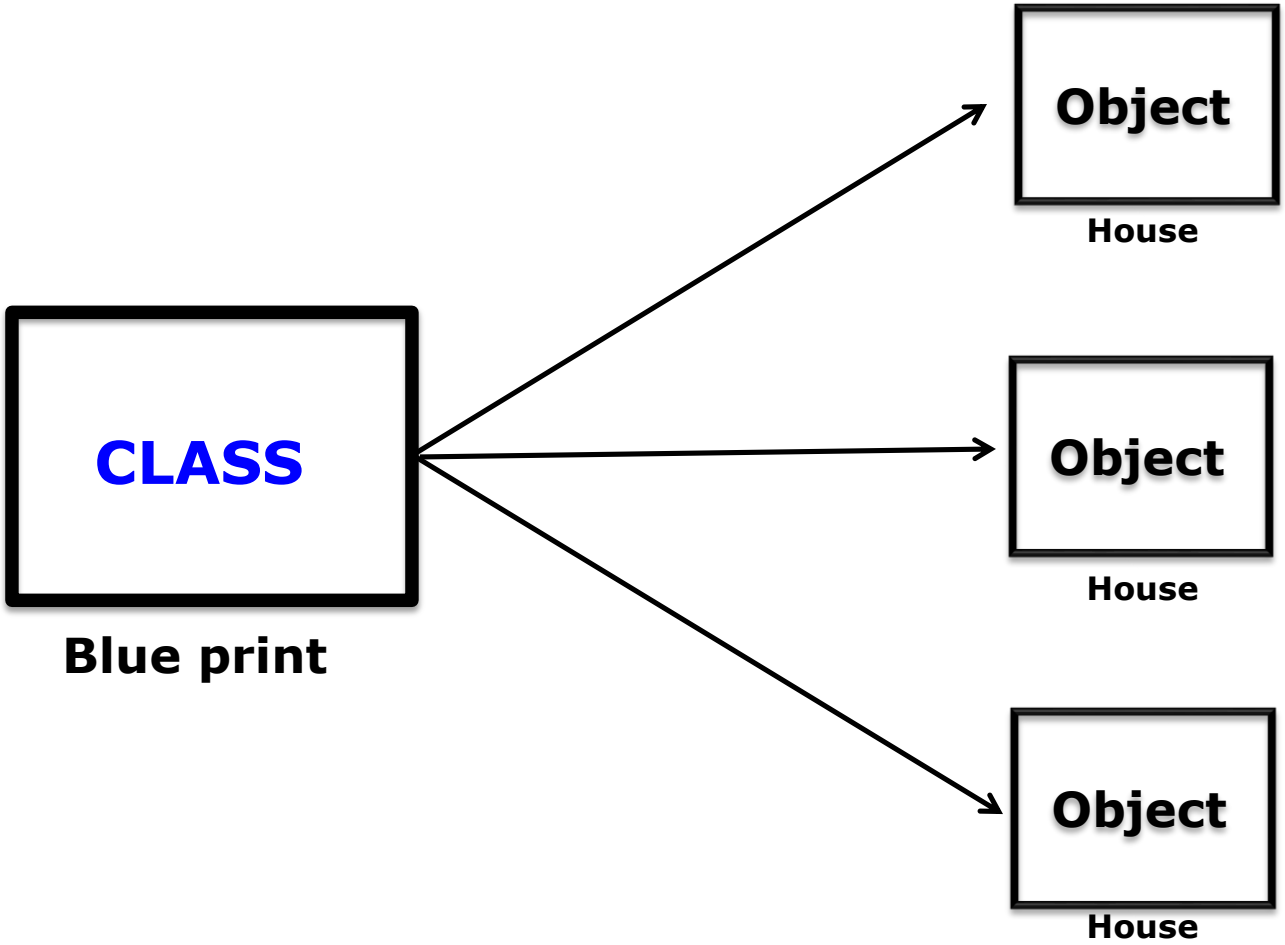
- Class
- Object

1. Turmeric powder - 100 gms
2. Sugar - 1 kg
3. Jaggery - 1/2 kg
4. Idli rice/Boiled rice/Salem rice - 5-7 kgs
5. Steamed rice or Raw rice/Sona masoori - 5-7 kgs
6. High quality raw rice for Pongal - 1 kg
7. Dosa rice (optional) - 2 kgs
8. Basmati rice - 1 to 2 kgs

BLUE PRINT


bedroom
13x12
bath
WH
mechanical
breakfast bar
kitchen
covered patio
13x13
living
15.5x13
©61custom

Class and Objects



C Language

```
int  id;

void display()
{
  printf(id);
}
```

Data member

Member function

```
class Student
{
    int id;
    void display()
    {
     cout<<id;
    }
}
```

```
class Student
{
    int id;
    void display()
    {
     cout<<id;
    }
}
```

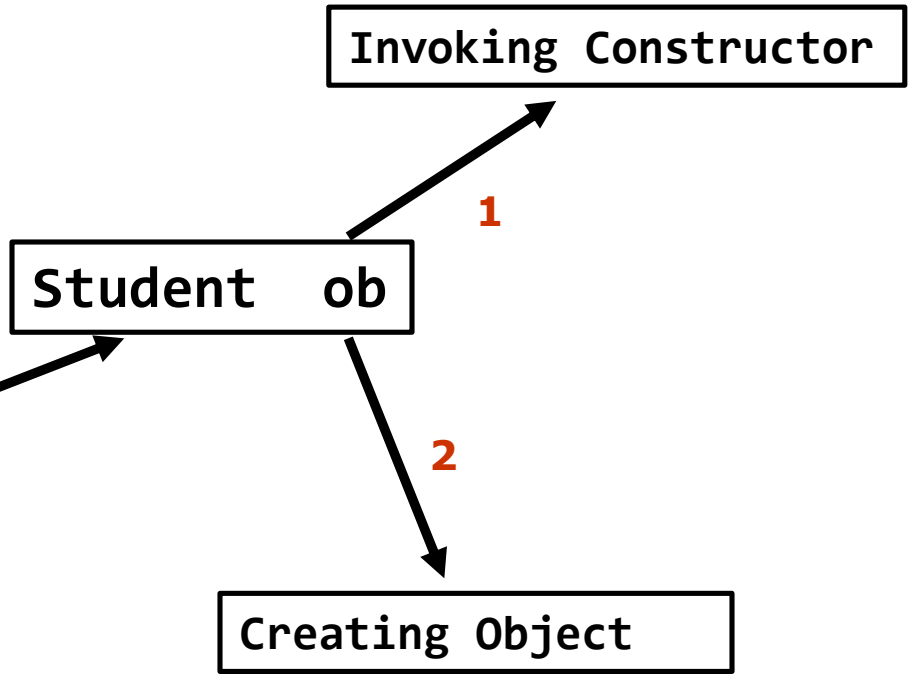
Blue Print

```
class Student
{
    int id;
    void display()
    {
      cout<<id;
    }
}
```

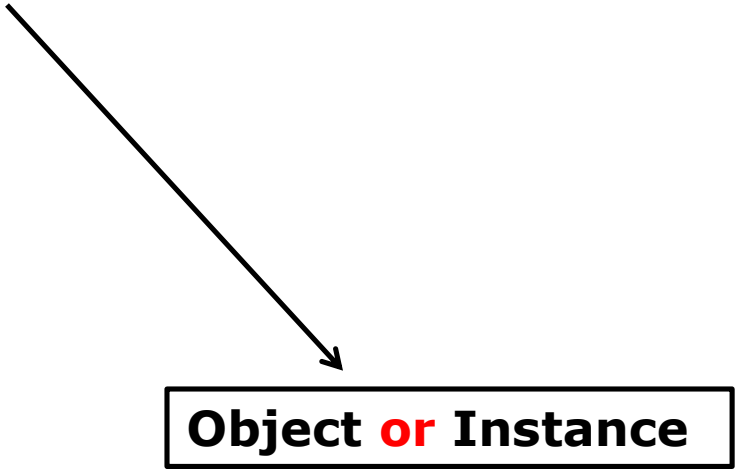
```
Student()
{
}
```

Blue Print

```
class Student
{
 int id;
  Student()
  {

  }
}
```



```
Student  ob;
```



```
int        x;
```

Pre-defined data type

```
Student  ob;
```

User-defined data type

```
class Demo
{
        int     x;
}
```

```
class Student
{
    int         rno;
    String      name;
}
```

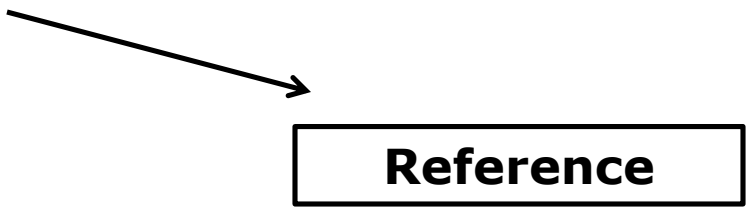
```
class String
{

}
```

```
Student  s1;
```

Object creation of Student class

Pointers

```
int        *x;
```




```
Student    *ob;
```

Pointers

```
Student  ob;
```

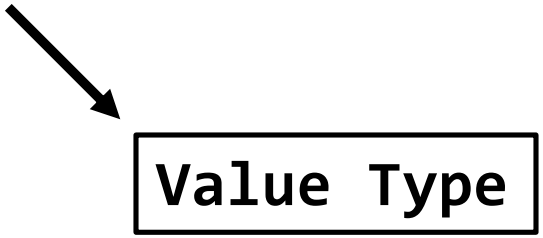
Object creation

```
new Student()
```

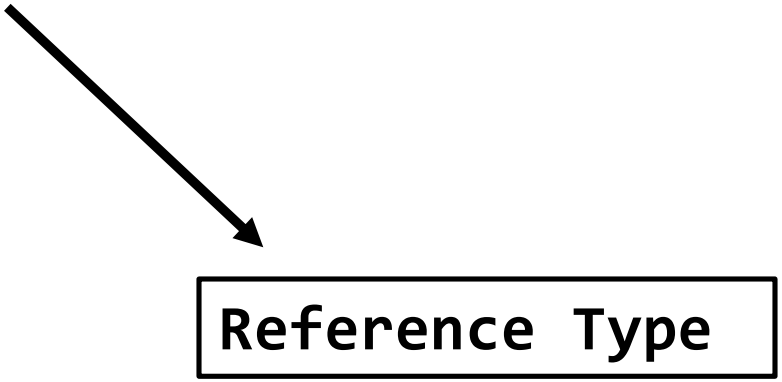
Object creation

Object Creation in C++

```
Student    ob;
```

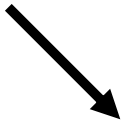



```
Student   *ptr = new Student();
```



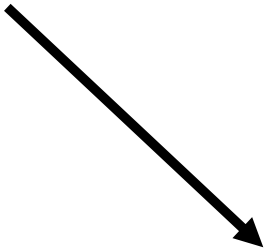
Object Creation in C++

```
Student    ob;
```



Value Type

```
Student   *ptr = &ob;
```



Reference Type

JAVA

```
Student   *ob1;
```

Reference creation in **C++**

```
Student    ob2;
```

Object creation in **C++**

Reference creation in **JAVA**

Reference

```
Student   *ob1;
```

Reference in C++

```
Student   ob2;
```

Reference in JAVA

C++

```
Student   *ob;
```

Reference

```
new Student()
```

Object creation

Java

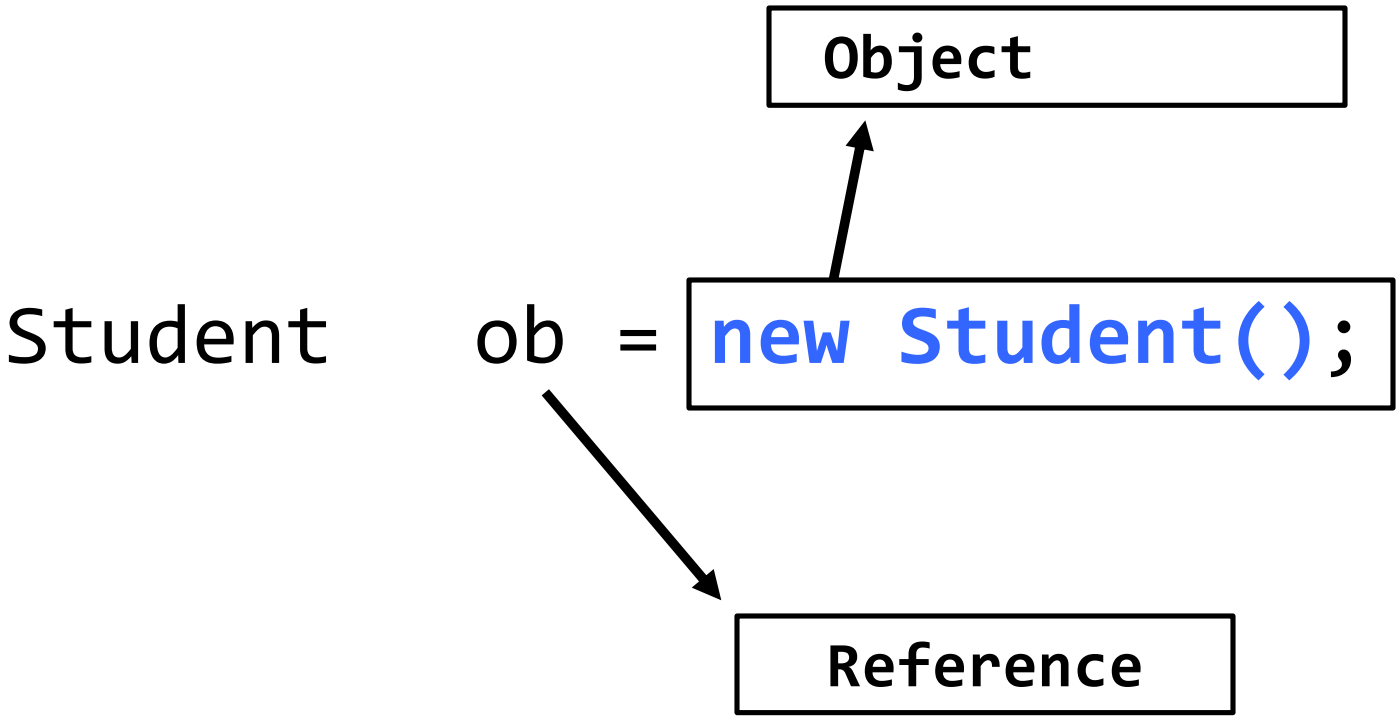
```
Student  ob;
```

Reference

```
new Student()
```

Object creation

Object Creation in Java



Object Creation

```
Student    ob = new Student();
```

Object

```
class Student
{
  int id;
  Student()
  {
  }
}
```

```
Student  ob = new Student();
```

Instance (or) Object

```
Student    ob  =   new Student();
           ob.id=200;
           ob.display();
```

Data Members

```
class Employee
{
  int        id  =  100;
  Address    ob =  new Address();
}
```

```
class Address
{
}
```

Class

```
class Student
{
    int id;
    void display()
    {
     cout<<id;
    }
}
```

Blue Print

Types of Variables and Methods

- **Instance variable and Instance Method (non-static).**
- **Class variable and Class Method (static).**
- **Local Variable**

```
class Demo
{
    static int a;
            int b;
}
```

Static Member

Non-Static Member

```
class Demo
{
    static int a;
            int b;
}
```

Static Member

Non-Static Member

JRE

JVM

Class loading

Object creation

JDK

Source code (.java)

Byte code (.class)

```
class Demo
{
    static int a;
           int b;
}
```

```
Demo.a = 5000;
```

```
Demo obj=new Demo();
obj.b=1000;
```

Instance variable and method

```
class Demo
{
    int a;
    void sum()
    {
     cout<<a;
    }
}
```

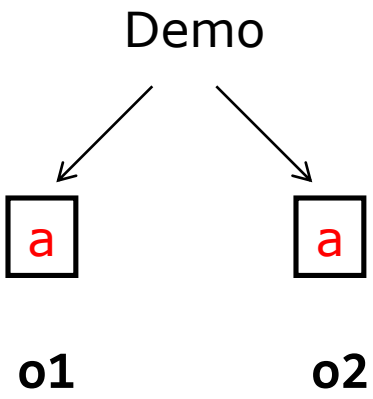
Instance Variable

Instance Method

Instance variable and method

```
class Demo
{
    int a;
    void sum()
    {
     cout<<a;
    }
}
```

```
Demo o1=new Demo();
o1.a=1000;
o1.sum();

Demo o2=new Demo();
o2.a=3000;
o2.sum();
```



Class variable and method

```
class Demo
{
    static int a;
    static void sum()
    {
     cout<<a;
    }
}
```

Class Variable

Class Method

Class variable and method

```
class Demo
{
    static int a;
    static void sum()
    {
     cout<<a;
    }
}
```

```
Demo.a = 5000;
Demo.sum();
```

Local Variable

```
class Demo
{
    void sum()
    {
        int a;
        cout<<a;
    }
}
```

Local Variable

Packages

```
package yahoo;

class Registration
{
}
```

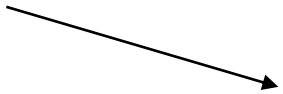
Sub packages

Classes

Interfaces

Java Packages

```
java.lang.*;
```



Object

System

Printing Statement

```
System.out.println("Welcome");
```

System Class

```
class System
{
    int x;
    int y;
}
```

```
System ob1 = new System();

ob1.x = 100;
ob1.y = 200;
```

System Class

```
class System
{
    static int x;
    static int y;
}
```

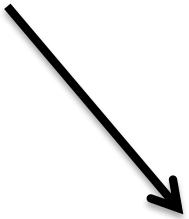
```
System.x = 100;
System.y = 100;
```

```
class PrintStream
{
      void println(int);
      void println(String);
}
```

System Class

```
class System
{
  static PrintStream out = new PrintStream();
}
```

```
PrintStream  out = new PrintStream();
```



Object of PrintStream class

Class

Variable (static)

```
System.out
```

Class

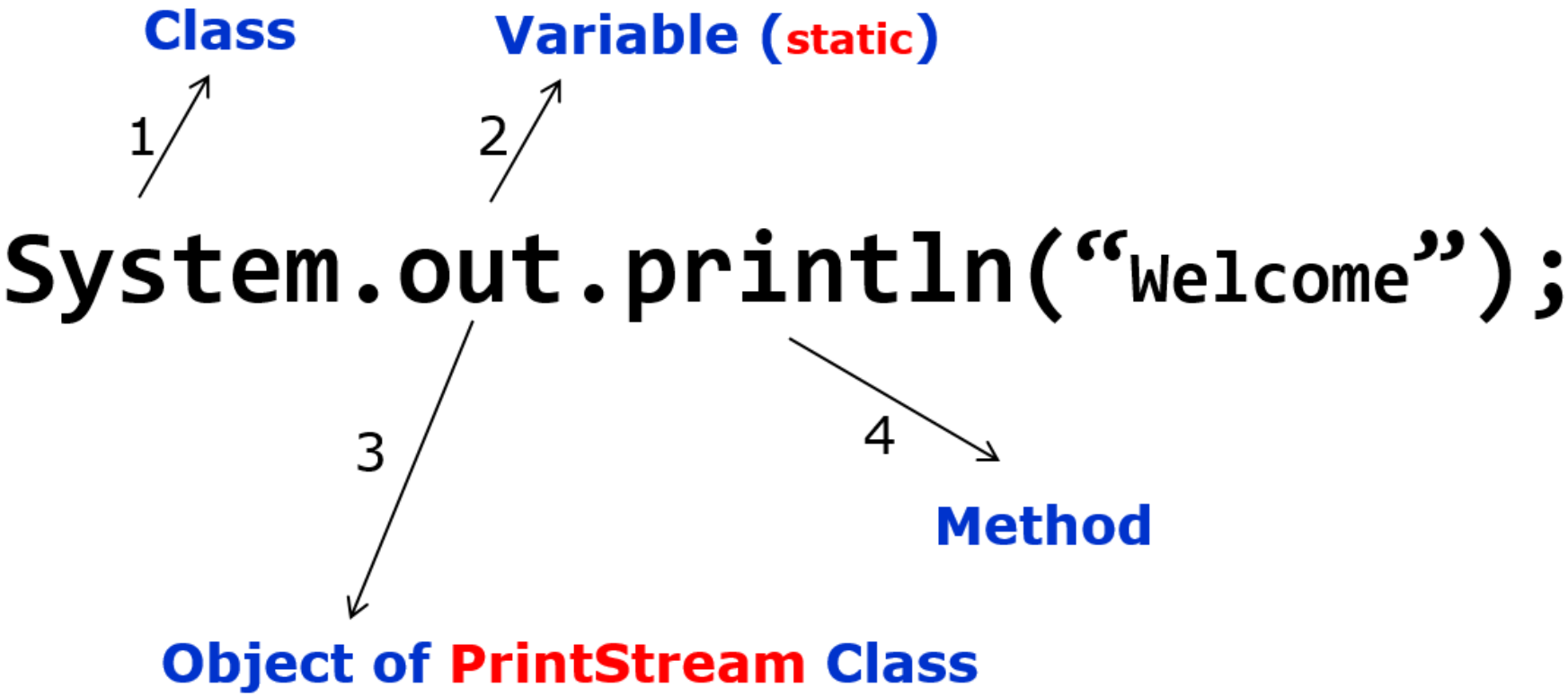
Variable (static)

```
System.out
```

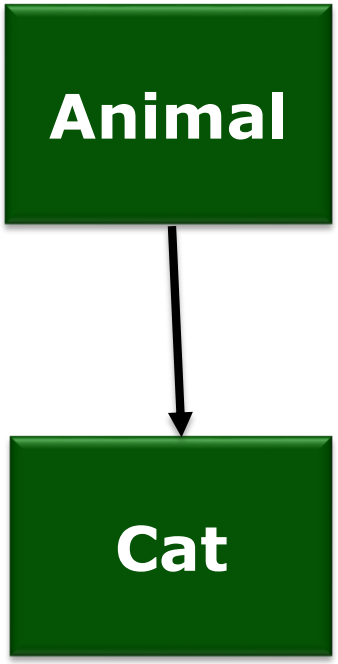
Object of PrintStream Class

PrintStream Class

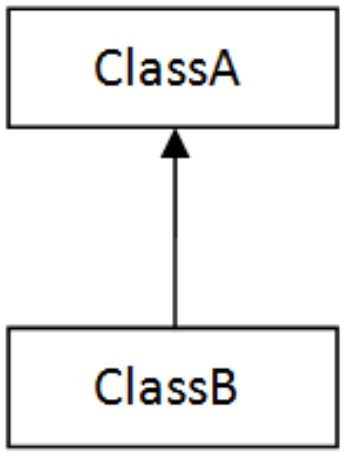
```
class PrintStream
{
      void println(int);
      void println(String);
}
```


Class
Variable (static)
1
2
System.out.println("Welcome");
3
4
Method
Object of PrintStream Class

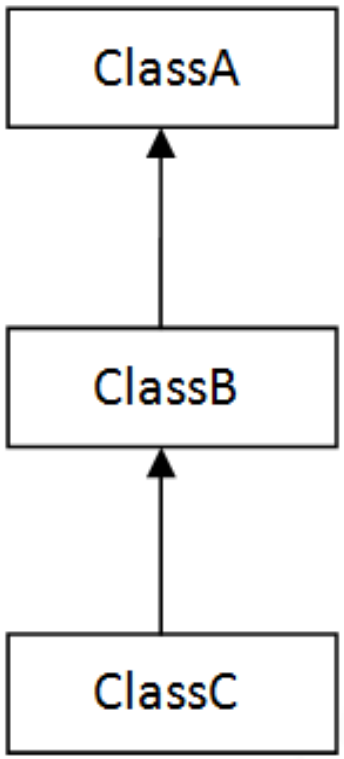
```
class Animal
{

}
class Cat extends Animal
{

}
```



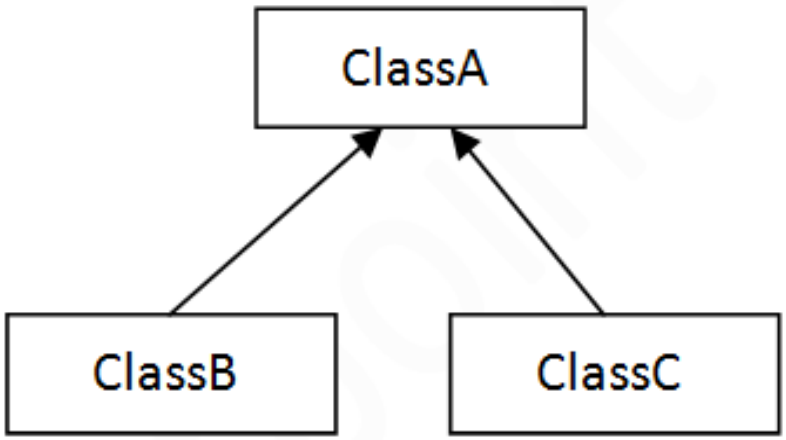
Inheritance



1) Single



2) Multilevel



3) Hierarchical

```java
class A{

    int x;
    void test(){
        System.out.println(" X  : "+x);
    }

}

class B extends A{

    int y;
    void show(){
        System.out.println(" X  : "+x+" Y  : "+y);
    }

}
```

Relationship

- Inheritance(IS-A)
- Aggregation(HAS-A)

```
class Employee{
        float salary;
}
class Programmer extends Employee{
        int bonus;
}
```

Programmer **IS-A** Employee

```
class Address
{
        String city,state,country;
}

class Employee
{
        int       eid;

        Address   ob1;
}
```

Employee **HAS-A** Address

```
class Address
{
        String city,state,country;
}

class Employee
{
        int       eid;

        Address  ob1 = new Address();
}
```

Employee **HAS-A** Address

```
class Demo
{

}
```

```
class Demo extends Object
{

}
```

```
import java.lang.*;

class Demo extends Object
{
    Demo()
    {
    }
}
```

Default Package

Default Base Class

Demo.java

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

Java Program Execution Steps

1. Type the Java Program in Notepad and save in any user directory

   eg. E:\Test\Demo.java

2. Go to Command Prompt and change the directory location

   eg. cd E:\Test

3. Set Path to Java Installed Directory

   E:\>Test> set path = c:\Program Files\Java\jdk1.8.0_111\bin

4. Compile and Run the Java Program

   E:\>Test> javac Demo.java

   E:\>Test> java Demo