Java Programming

- Why Java?
- History of Java
- Java Environment
- Java Installation on Windows
- Running our First Java Program

Java Language Features

- Simple.
- Object-Oriented.
- Portable.
- Platform independent.
- Secured.
- Robust.
- Architecture neutral.
- Interpreted.

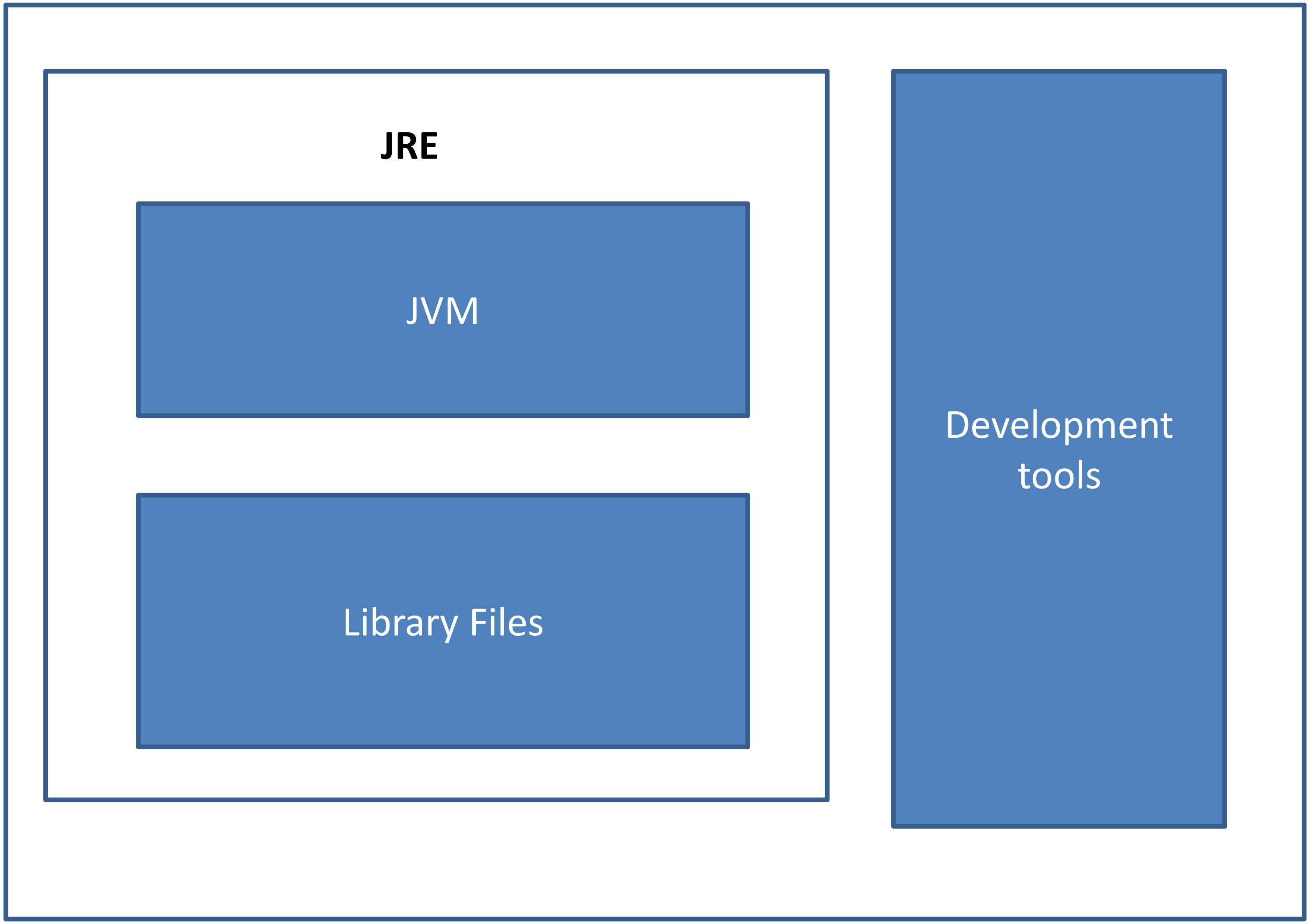
Java History

James Arthur Gosling, OC (born May 19, 1955) is a Canadian computer scientist, best known as the founder and lead designer behind the Java programming language.
He is generally credited with having invented the Java programming language in 1994. He created the original design of Java and implemented the language's original compiler and virtual machine

Java Environment

- **JDK** (Java SE Development Kit) - For Java Developers. Includes a complete **JRE** plus tools for developing, debugging, and monitoring Java applications.

JDK



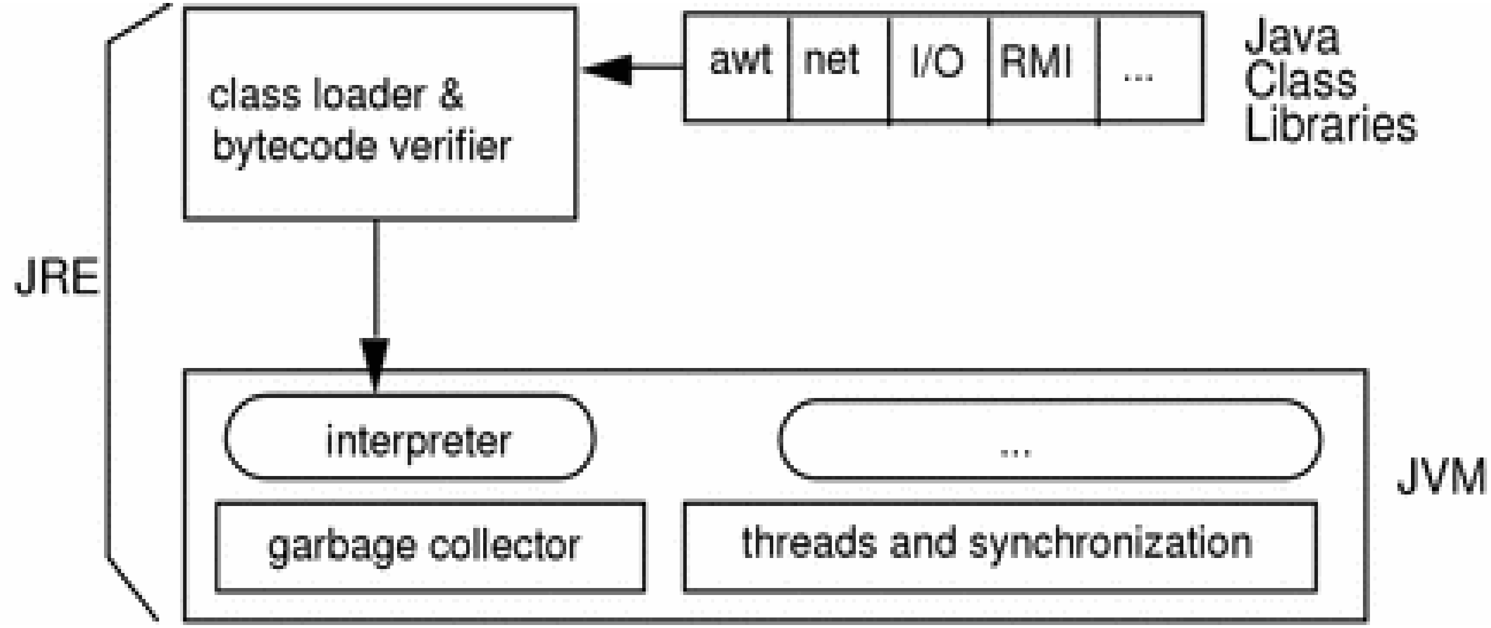
JRE

JVM

Library Files

Development tools

JRE


class loader &
bytecode verifier
awt
net
I/O
RMI
...
Java
Class
Libraries
JRE
interpreter
...
JVM
garbage collector
threads and synchronization

JDK Versions

Version	Release date
JDK Beta	1995
JDK 1.0	January 1996
JDK 1.1	February 1997
J2SE 1.2	December 1998
J2SE 1.3	May 2000
J2SE 1.4	February 2002
J2SE 5.0	September 2004
Java SE 6	December 2006
Java SE 7	July 2011
Java SE 8 (LTS)	March 2014
Java SE 9	September 2017
Java SE 10 (18.3)	March 2018
Java SE 11 (18.9 LTS)	September 2018
Java SE 12	March 2019

Installation

- Go this link for the latest JDK download

https://www.oracle.com/technetwork/java/javase/downloads/index.html

Running our first Program

- Create a Folder to store all your programs

IDE for Java

- **NetBeans**
- **Eclipse**
- **IntelliJ IDEA Community Edition**
- **Android Studio**
- **Enide Studio 2014**
- **BlueJ**
- **jEdit**
- **jGRASP**
- **JSource**
- **JDeveloper**
- **DrJava**

Downloading Eclipse

- https://netbeans.org/downloads/8.0.2/
- https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers

Simple Java Program Structure

```
public class first
{

    public static void main(String[] args)
    {

        System.out.println(“All the best");

    }

}
```

Java Data Types

Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Operators in Java

Operator Type	Category	Precedence
Unary	postfix	expr++ expr--
	prefix	++expr --expr +expr -expr ~ !
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Reading Input from User

```
import java.util.*;
public class first {

public static void main(String[] args) {

System.out.println("Enter your name");
Scanner test= new Scanner(System.in);
String name=test.nextLine();
System.out.println(name);

}

}
```

If else Statement

- If
- If else
- If else ladder
- Nested if else statement

Syntax - if

```
 if(condition)
{
   //code to be executed
 }
```

Syntax – if else

```
 if(condition)
{
   //Statements that will be executed if condition is true
}else
{
   //Statements that will be executed if condition is false
}
```

Nested If else

```
if(condition1)
  {
  //execute statements if condition1 is true
  }else if(condition2){
  //statements to be executed if condition2 is true
  }
  else if(condition3){
  //statements to be executed if condition3 is true
  }
  ...
  else{
  //statements to be executed if all the conditions are false
  }
```

Switch Statement Syntax

```
switch(expression)
{
    case value1:
     //statements;
     break;
    case value2:
     //statements;
     break;
    ......

    default:
     Code to be executed
    Break;
}
```

Loops in Java

- For
- While
- Do while

Syntax For Loop

```
for(initialization;condition;incr/decr)
{
   //statements
}
```

Loops in Java

```
public class hello {

public static void main(String[] args) {

for(int i=0;i<5;i++)
{
System.out.println("Hello");
}

}

}
```

While loop Syntax

```
 while(condition)
{
      //statements
}
```

While Loop

```
public class hello {

public static void main(String[] args) {

int i=0;
while(i<5)
{
System.out.println("Hello");
i++;
}

}

}
```

Do While Loop Syntax

```
 do
{
   //statements

}while(condition);
```

Do while loop

```
public class hello {

public static void main(String[] args)
{

    int i = 0;
    do
    {
        System.out.println("Hello");
        i++;
    } while (i < 5);

}

}
```

Break and Continue

Exercise

- Check if a given number is odd or even number [using if statement as well as switch case]
- BMI Calculation (bmi = kg/m^2)
  (ie 60/1.68^2) [60 kg, 168 cm]
- Print the sum of first n numbers. If N is 3 then print the sum of 1+2+3 to the user. Get n from the user
- Print grade based on absolute marking

Exercises

- Read the radius and print the area of a circle
- Read year and check if the given year is a leap year
- Display Subject Name based on room number. If the user enters 823 then display Java Programming , If the user enters 824 then display Python programming for any other input display Invalid input to the user
- Print the sum of the series 1^2+2^2+3^2 up to n terms
- Print the multiplication table by getting the n from the user.
- Provide the option of adding two numbers to the user until the user wants to exit

Nested Loops

- Print this pattern for n lines

```
*

**

***

****
```

Print this pattern for n lines

```
1234
123
12
1
```

Print this pattern

```
1
12
123
1234
1234
123
12
1
```

Arrays in Java

- An array is a container that holds data (values) **of one single type**.
- For example, you can create an array that can hold 100 values of int type.

Array Declaration

```
int[] arrayname = new int[5];
```

Initializing an array

```
int[] test = {1,2,3,4,5};
```

Perform the following Exercises

- Sort an array of element using bubble sort
- Remove duplicate elements from a sorted array
- Reverse the contents inside an array
- Search for an element inside the array using linear search

Multidimensional arrays in Java

```
int[][] test = {{1,1},{2,2}};
```

Perform the following exercises

- Check if a given input matrix from a user is an identity matrix
- Display the transpose of a given matrix
- Display the sum of rows in a matrix
- Display the addition result of two matrices