**Basic Java**

**What is Java?**

**Ans:** Java is a popular programming language created in 1995. It is own by Oracle. More than 300 billion devices run Java.

**Features of Java?**

**Ans:** There are nearly 12 features, which makes Java a popular and dependable language all over the world. Those are given below,

1. **Simple**: Java is a simple language because it syntax is simple, easy, clean and easy to understand. Its syntax is based on C++ so, it’s easier to learn java who knows about the C++ programming language. It also removed many complicated and rarely used feature like pointers and operator overloading. Java also contain a feature called automated garbage collection, which removes the unreferenced objects.
2. **Object oriented**: Java is an object-oriented programming. Everything in java is an object. It simplify the software development and maintenance by providing some rules and combining different types of objects that incorporates both data and behavior. Basic object oriented concept are Inheritance, polymorphism, abstraction, encapsulation.
3. **Platform independent**: Java is platform independence language because it compiled be the compiler into a byte-code. This byte-code can be run from different platform like windows, Linux, Mac/Os etc. So, Java defines as write once and run anywhere(WORA)
4. **Secure**: Java is known the best for its security. We can develop a virus free system by Java. There are some reasons -> no explicit pointers and java runs inside a virtual machine sandbox. It adds security by separating the package for the classes of the local file system from those that are imported from network sources. It checks the code fragments for illegal code that can violate access right to objects.
5. **Robust**: Robust means strong. Java is a robust language because it uses strong memory managements; avoid using pointers, which keeps the program secure, automated garbage collection, which helps to remove unused objects. There is exception handler feature which makes java a robust language.
6. **Architecture-neutral:** Java is architectural neutral because there is no implementation dependency. In java for both 32 and 64-bit architecture Int data type occupies 4 bytes of memory.
7. **Portable:** Java is called portable because it byte code can be carry anywhere and can run without any implementation.
8. **High Performance:** Java is faster than any other interpreted language because Java byte code is close to native code.
9. **Distributed:** There is RMI and EJB facilities in java that creates distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.
10. **Multithreaded:** Java programs can deal with many task in at once by using multithread. The main benefit is it does not occupy memory for each thread shares common memory area. Threads are important for multimedia and web applications.
11. **Dynamic:** Java is dynamic language. Because it load classes on demand. It supports the functions of its native classes also C,C++.

**C++ vs Java?**

**Ans:** The differences are given bellow

|  |  |
| --- | --- |
| Java | C++ |
| Platform independent | Platform depended |
| Dose not support goto statement | Support goto statement |
| Does not support multiple inheritance through classes. By using interface it can be achieved | Support multiple inheritance |
| Java avoid pointer and operator overloading. | C++ supports pointer and operator overloading. |
| Java is compiler and interpreted language | C++ is compiler language |
| Java supports only call by value | C++ supports call by value and call by reference. |
| Java has built in thread support | C++ has no built in thread supports. Thread support can be avail by third party libraries. |
| Java is object oriented language | C++ is object oriented as well as procedural. |

**Create Java see output Hello World Program?**

**Ans:** To create Java program we have to install jdk for creating the environment.

The first program is –

class Simple{

public static void main(String args[]){

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

}

}

First we have to create a class . So we create simple class . For execute program we have to declare a public method. Because this method tell from where the program will start execution. Then by the support of another class System.out.println() we got the output. Hello world.

**What happen internally of the hello world program?**

**Ans:** The written java code converted to a byte-code.

Byte-code

Compile

Java code

Then execute the program. There also create a file on named .class which contains the byte code.

After that with the help of hardware, this shows the output.

**Difference between JVM, JRE and JDK?**

**Ans:** The difference are given below

1. JVM (Java Virtual Machine) : It is a specification that provides a runtime environment in which Java bytecode can be executed. It has no physical existence. JVM performs the following main tasks Load codes, Verify codes, Execute codes, provides run time environment.
2. JRE(Java Runtime Environment) : The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.
3. JDK(Java Development Kit): JDK is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

**Information about Java variable?**

**Ans:**  Variable means a container which hold value. There are 3 types of container in Java

Local variable: The variable which are declared inside a method, called local variable.

Instance: The variable which are declared in a class but outside the method is called instance variable.

Static Variable: The variable which are declared as static is called static variable. It can be local. Static variable can be declared once and share among the instances of the classes.

Example :

class Basic{

int number = 50; //instance variable

static int number1=100;//static variable

void method(){

int n=90;//local variable

}

}

**About Data type of Java?**

**Ans:**  There two types of data type in Java

1. Primitive Data Type (int,float,double,char etc)
2. Non- Primitive Data Type (Sring,classes,interfaces,arrys etc)

**About Unicode in Java?**

**Ans:** Unicode is a universal international standard character encoding system that is capable of describing most of the written languages of the world.

1. ASCII (American Standard Code for Information Interchange) for the United States.
2. ISO 8859-1 for Western European Language.
3. KOI-8 for Russian.
4. GB18030 and BIG-5 for chinese, and so on.

Unicode is initiated because a particular part of the code is written in different letters for different languages all over the world. Some common shorts of code are encoded into single byte but others needs two or more bytes. So solve this type of problem there have been introduce a common language standard which is called Unicode.

Highest value : u0000

Lowest value : uFFFF

**About operator in Java?**

**Ans:**  Operators means the symbol which is used to perform operations.

|  |  |  |
| --- | --- | --- |
| Operator type | Category | Symbol |
| Unary | Postfix | i++,,i-- |
| prefix | ++I,--i |
| Arithmetic | Multiplicative | \*, / , % |
| additive | + , - |
| Shift | shift | >>, << , >>> |
| Relational | Comparison | > , < , >=, instanceof |
| Equal | == , != |

|  |  |  |
| --- | --- | --- |
| Bitwise | Bitwise AND | & |
| Bitwise exclusive OR | ^ |
| Bitwise inclusive OR | | |
| Logical | Logical AND | && |
| Logical OR | || |
| Ternary | Ternary | ? : |
| Assignment | Assignment | =, += ,-=, \*= ,/= ,%= ,&= ,^= ,|=, <<= ,>>= ,>>>= |

**About some string method of Java?**

**Ans:**  The brief of some methods are given below.

charAt() -> Returns the character at the specified index (position);

compareTo() -> Compares two strings letter wise.

concat() -> Appends a string to the end of another string.

contains()-> Checks whether a string contains a sequence of characters.

copyValueOf() -> Returns a String that represents the characters of the character array.

endsWith()-> Checks whether a string ends with the specified character(s)

equals()-> Compares two strings. Returns true if the strings are equal, and false if not

getBytes() -> Encodes this String into a sequence of bytes using the named charset, storing the result into a new byte array

getChars() -> Copies characters from a string to an array of chars

hashCode() -> Returns the hash code of a string

indexOf() -> Returns the position of the first found occurrence of specified characters in a string

isEmpty() -> Checks whether a string is empty or not

length() -> Returns the length of a specified string

matches() -> Searches a string for a match against a regular expression, and returns the matches

replace() -> Searches a string for a specified value, and returns a new string where the specified values are replaced

replaceAll() -> Replaces each substring of this string that matches the given regular expression with the given replacement

split() -> Splits a string into an array of substrings

startsWith() -> Checks whether a string starts with specified characters

toCharArray() -> Converts this string to a new character array

toLowerCase() -> Converts a string to lower case letters

toString() -> toStringReturns the value of a String object

toUpperCase() -> Converts a string to upper case letters

trim() -> Removes whitespace from both ends of a string

valueOf() -> Returns the primitive value of a String object

**About arrays in Java?**

**Ans:** Array is collection of variable of same data type.

Declaration of array ->

**Int[ ]** variable name = new **int[10];**

**String[ ]** variable name = new **String[10];**

How to take input of an array ->

Scanner input = new Scanner(System.in);

**Int[ ]** num = new **int[10];**

For (int i =0 ; i< 10 ; i++) {

Num[ ] = input.nextInt();

}

Sorting an array ->

Arrays.sort(number);

Then print the whole value;

**About ArrayList in Java?**

**Ans:**  The main difference between array and array list is array is static and arrayList is dynamic.

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| Not Resizable | Resizable |
| Fast | Slow |
| Array.Length method is used to find out the array size | Array.size method is used to find out the array size |

How to declare Arraylist ->

ArrayList<integer> number = new ArrayList<integer> (); [No need to declare any array size]

number.add(9); [to add number ]

number.add(10); [to add number on array list ]

number.add(4,9); [ the first value is for index and the second one is for value ]

number.remove(2); [ inside the box 2 indicate the index number ]

number.removeAll(); [ it will clear the arraylist ]

number.contains(30); [it will indicate the value inside the bracket is in the list or not]

number.indexOf(40); [it will return the value is in which index]

number.set(4, 50); [it will set the 4th index value 50 and remove previous value]

number.get(4); [it will return the value of 4th index];

to sort the array we have to use Collections.sort(number); [ascending order]

Collections.sort(number, Collections.reverseOrder()); [sort the list in desending Order]

**Use of Final keyword?**

Ans:

if one variable declared with final keyword then it can’t be change

If a method declared with final keyword, it can’t be override

It a class declared with final keyword then class cant be inherit.