**JAVA**

* Features of JAVA

          .Object oriented

        . Platform independent

          .  Portable

          . Dynamic

          . Robust and Secure

* Modifiers

1. Access modifiers

* Default – Visible to the package
* Private- Visible to class
* Public- Visible to world
* Protected- Visible to package and all subclass

(b)Non Access modifiers

* Static modifiers- For creating class methods and variables
* Final modifiers- For finalizing implementation of classes and methods
* Abstract modifiers- For creating abstract classes and methods
* Datatypes of JAVA

1. Primitive Datatype

This specifies size and type of variable values and have no additional method.

Eg: byte, long, short, int

1. Non Primitive Datatype

These are the reference types because they refer to objects.

Eg: arrays, classes, interface

* If Statement in JAVA – This is the most simple decision making statement. It is used to decide whether a certain block of statements will be executed or not.
* If-Else Statement in Java – If the value if false, then the else statement is executed.
* Switch Statement in Java- This is really a multiple if-else  condition it can jump into case value and execute. But using of 'DEFAULT' is not mandatory in  this. It is better using 'break' statement otherwise code executes further.
* Array- This is a collection of data structures contains group of elements. It may be an integer type or string type.
* Constructor - Similar to a method that is invoked when an object of class is created. They are not members of the class. It is automatically called up when a new instance of object is created.
* Static Variable-  The static variable in JAVA is belongs to the class and initialized only once at the start of execution. Simply it belongs to the class and not to the object.
* This Operator   - Is a reference variable which point to the current object. This is used in instance variable passed as an argument with constructor also.
* Super Keyword - Is a reference variable which is used to refer immediate parent class object.
* POLYMORPHISM- (a) Method Overloading  - Is a compile time polymorphism which occurs within the class. It may or may not require inheritance.   If a class has multiple methods  having same name but different parameters, it is method overloading.                                                                                                                                                                                                                                                                                                                  (b)Method Overriding - Is a run time polymorphism performed in two classes with inheritance. If a subclass has same method as declared in parent class, it is method overriding.
* INTERFACE -    Is a mechanism to achieve abstraction.An interface is not extended by a class but it is implemented by a class.An interface can extend multiple interfaces.