Pre-Req:

Basic Manual Testing Concepts:

* Different types of manual Testing
* Regression Testing
* Test case
* Test Plan
* Bug life cycle
* Testing Methodologies
  + **Agile**
  + Verification and Validation

**Java**

* Basic components of Java
  + JDK
  + JRE
  + JVM
  + VERSIONS OF JAVA
* Basics of Java
  + OOPS Concept
  + class
  + interface
  + packages
  + constructors
* Variables
  + declare a variable
  + initialize a variable
  + Types of Variables
    - Local Variables
    - Global Variables
      * static variables
      * instance variables
      * constants
* Data types
  + primitive Data types
  + derived Data types
* Access Specifiers
  + private
  + default
  + protected
  + public
* Access Modifiers
  + static
  + abstract
  + final
  + synchronized
* Looping
  + for
  + while
  + do,while
  + foreach
* Conditional statements
  + if
  + if, else
  + if, elseif , else
  + switch
* String (java.lang)
  + charAt
  + split
  + indexOf
  + lastindexOf
  + toUpperCase
  + toLowerCase
  + equals
  + equalsIgnoreCase
* Exception Handling
  + try
  + catch
  + throw
  + throws
  + finally
  + User Defined Exceptions
* File Handling
  + Read file (properties file, text file, excel file)
  + Write into a file (properties file, text file, excel file)
* Arrays, Enumerator
* Collections
  + List
    - ArrayList
    - LinkedList
    - Vector
  + set
    - HashSet
    - TreeSet
    - LinkedHashSet
  + Map
    - HashMap
    - TreeMap
    - LinkedHashMap
* Generics

**Selenium:**

* What is Regression Testing
* What are the drawbacks of Manual Testing
* Why Automation testing is required
* Why Selenium????
* History of Selenium
* Selenium IDE
  + Beginners
  + Addon FF and Chrome
* ~~Selenium RC~~
* Selenium WebDriver / Selenium2.0 / Selenium3.0 / Alpha Selenium4.0
  + type
  + click
  + mouse movement
  + double click
  + right click
  + browser back / forward / refresh / maximize
  + read values from application
  + read Webtable
  + Auto suggestions
  + Automate tooltip info
  + Handle different / multi browser /tabbed browser
  + Calendar
  + Images
  + Popup
    - alerts
    - confirmation popup
    - hidden division popup
    - authentication popup
    - file download popup
    - file upload popup
  + Load the browser with customized settings
  + Execute Test on Different Browsers – IE, Chrome, FF, Opera, Safari
  + Data driven testing
    - excel
    - properties
  + End to end tests
* Selenium Grid
* Cucumber – BDD Approach
* Frame Work
  + Page Object Model (POM)
  + TestNG
    - annotations
    - validating HTML Reports
    - grouping the tests
    - Data driven testing
* Maven – Build Automation Tool
  + Maven Repository
    - Local Repository
    - Global Repository
  + POM.xml
  + setup a maven project
  + pre settings to be updated
  + Maven Build Life cycle phases
  + Execute the test without opening eclipse – Using Batch file
* GIT – Version Control Tool
* Jenkins – CI/CD Tool
  + how to configure Jenkins
  + how to create builds
  + how to execute tests in Jenkins
  + how to execute tests daily

====================================================================================================JAVA==========================================================================================================

1. Installation
2. Set the environment Variables
   1. Set JAVA\_HOME
   2. Update PATH Environment Variable



c. Open Command prompt and verify java



1. Components of Java



* JDK 🡪 Java development KIT
  + Compile the Java Program
* JRE 🡪 Java Runtime Environment
  + Run the Java Program
* JVM 🡪 Java Virtual Machine
  + Loads code
  + Verifies the code
  + Executes the code
  + Provides runtime environment
* History:

1. Editor – Eclipse, netbeans, intelliJ
2. HelloWorld JAVA Program
3. Understand the Java Program in detail.
   1. 

b. 

1. Data types
   1. data types

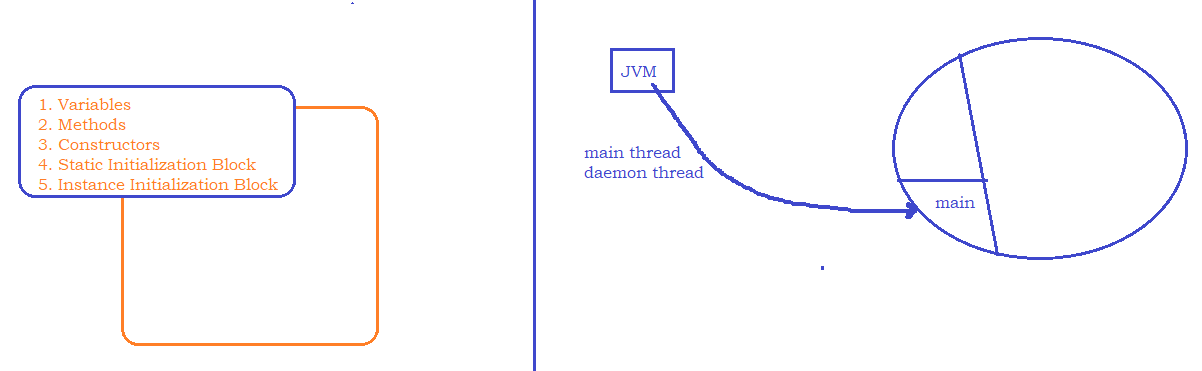


b. 

1. Packages
   1. always package should start com / org
   2. followed by company name / client name
   3. followed by project name
   4. followed by module / sub-module name
   5. ****
2. Variables
   1. declare 🡪 data\_type variable\_name
   2. initialize 🡪 variable\_name = Value;
   3. Types

****

**Members of Class : Execution by main thread**

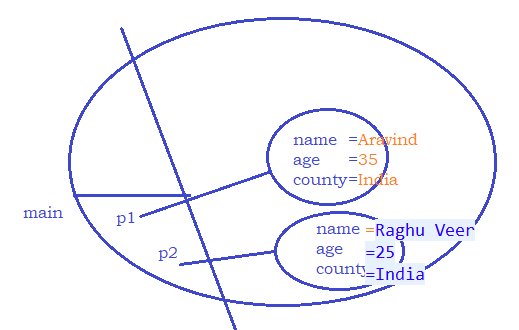
****

**Access members of one class inside another –**

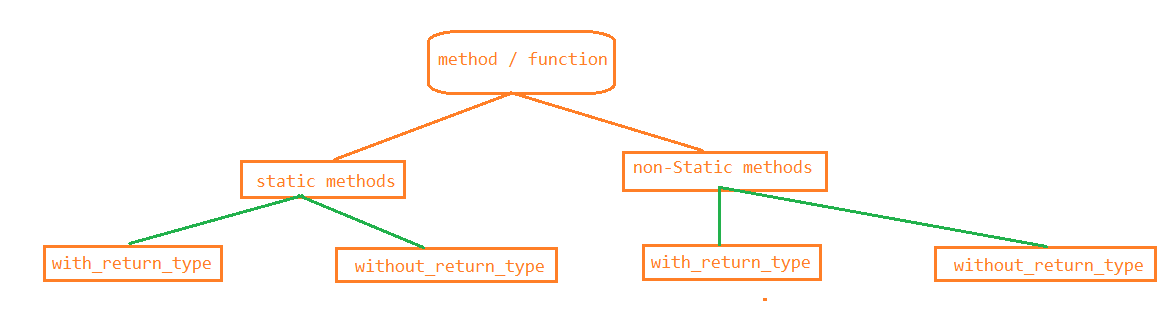
1. Create Object to the Class

2. Use dot operator to access the member of a class

Instance Variables



1. **Methods – Describes the behaviour of a class**

****

**[Access\_specifier] [Access\_modifier] return\_type name\_of\_method(ArgumentList )**

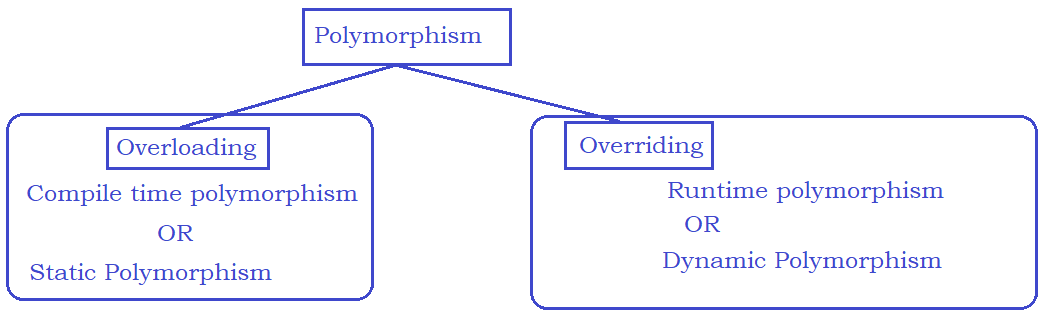
**{**

**--------------------**

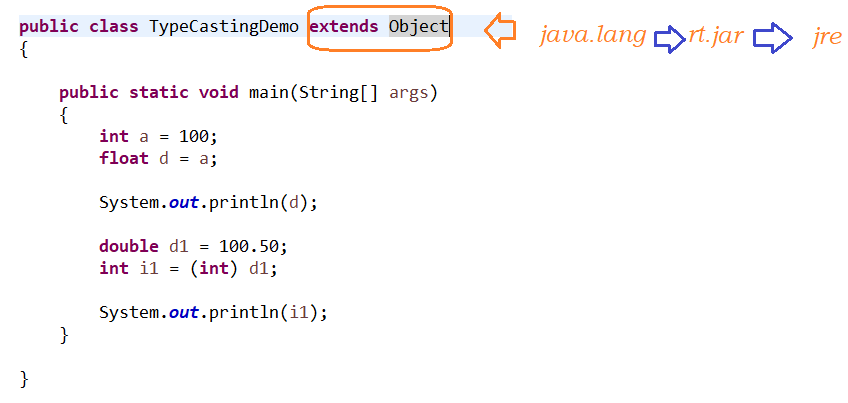
**--------------------**

**}**

* 1. **Variable number of Arguments to a function**
  2. **Polymorphism**

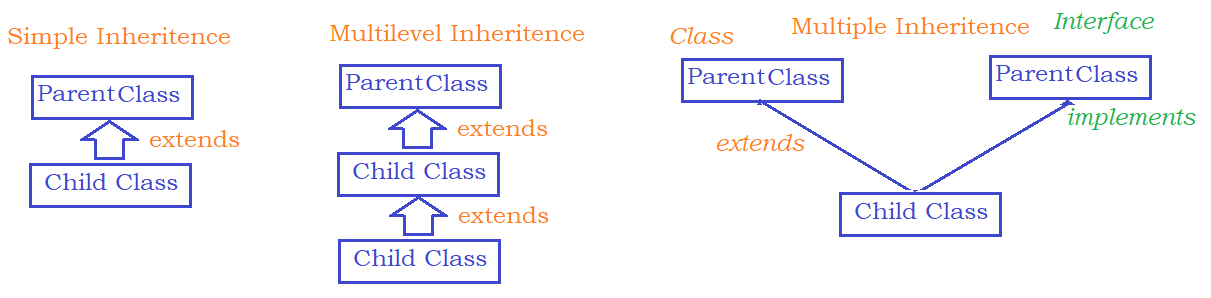
****

1. **Type Casting**

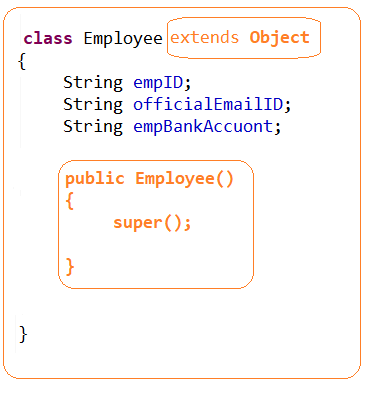
****

* 1. Auto type casting -> int to double, byte to int
  2. Explicit type casting -> double to int, int to byte

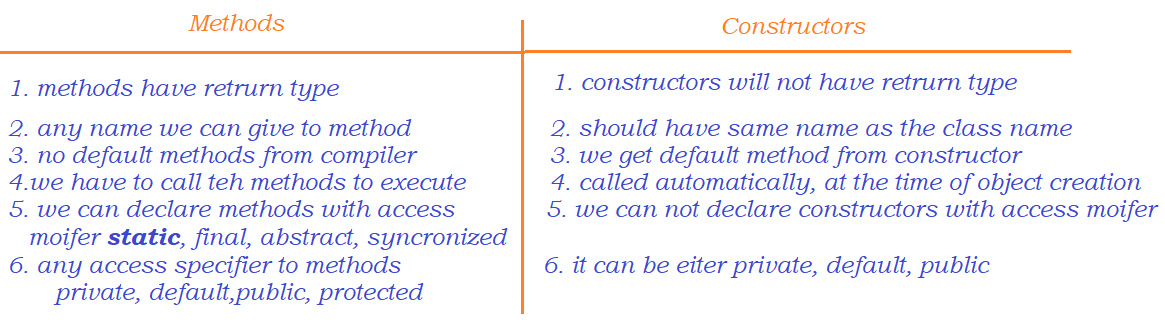
1. **Inheritance**

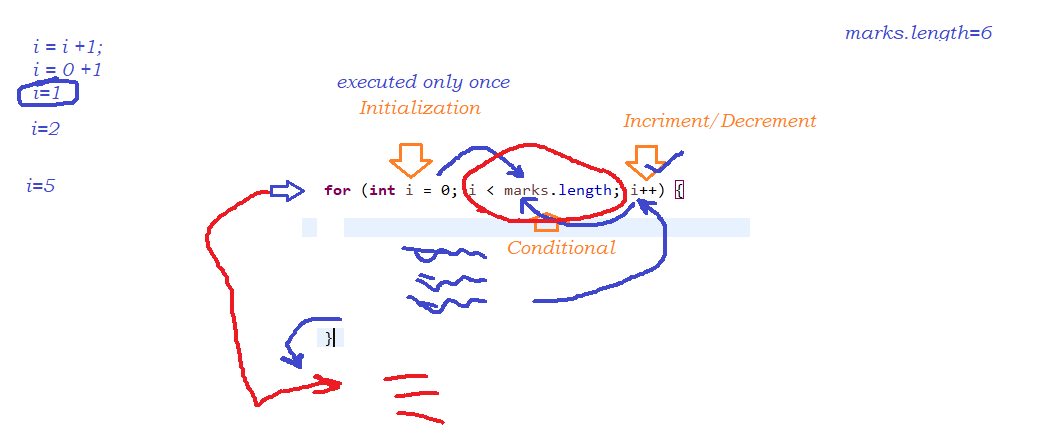
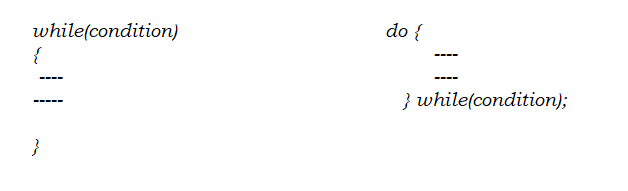
****

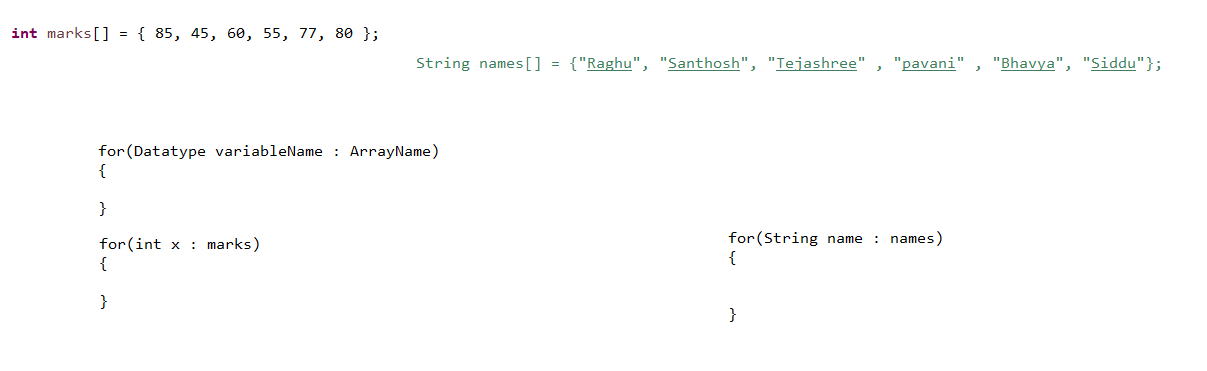
1. **Constructor**
   1. **Default Constructor**

****

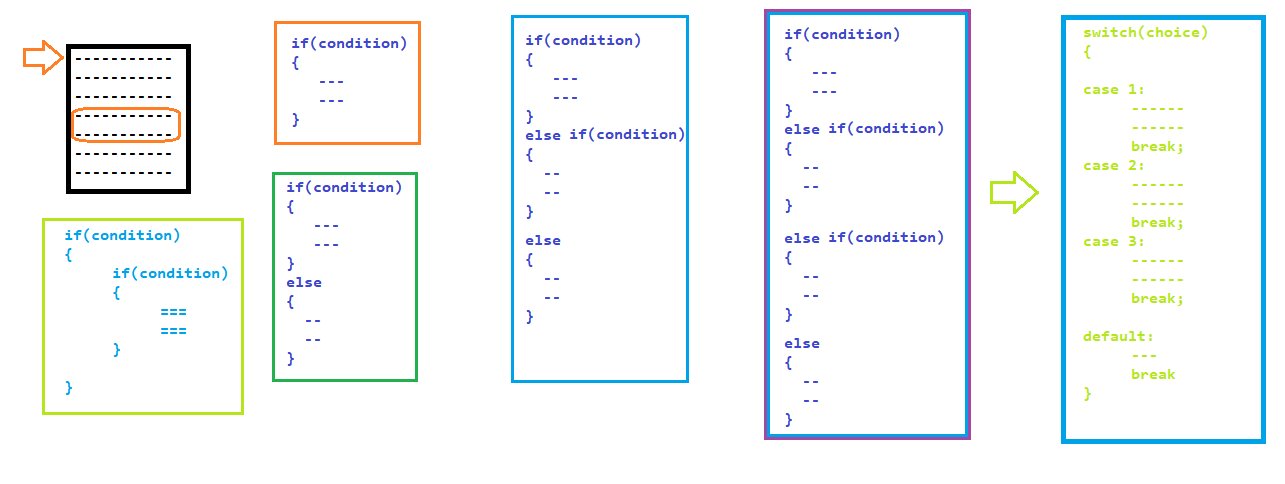
* 1. **Parameterized Constructor**
  2. **Difference between Methods and constructors**



1. **Arrays**
   1. single dimensional
   2. Multidimensional array
2. **Looping Statements**
   1. for
      1. ****
      2. ****
   2. while
      1. ****
   3. do,while
   4. foreach

****

1. Conditional Statements



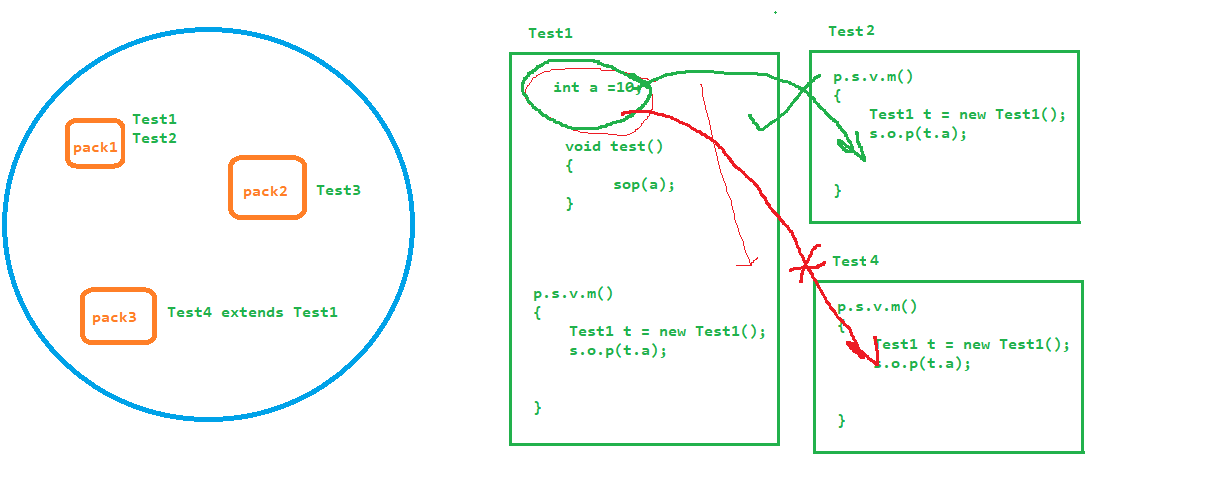
1. SIB and IIB
   1. Static initialization Block – will be called before executing main method
   2. Instance initialization Block – will be called before calling a constructor
2. **Abstract Class**

Class can be declared as abstract in 2 scenarios

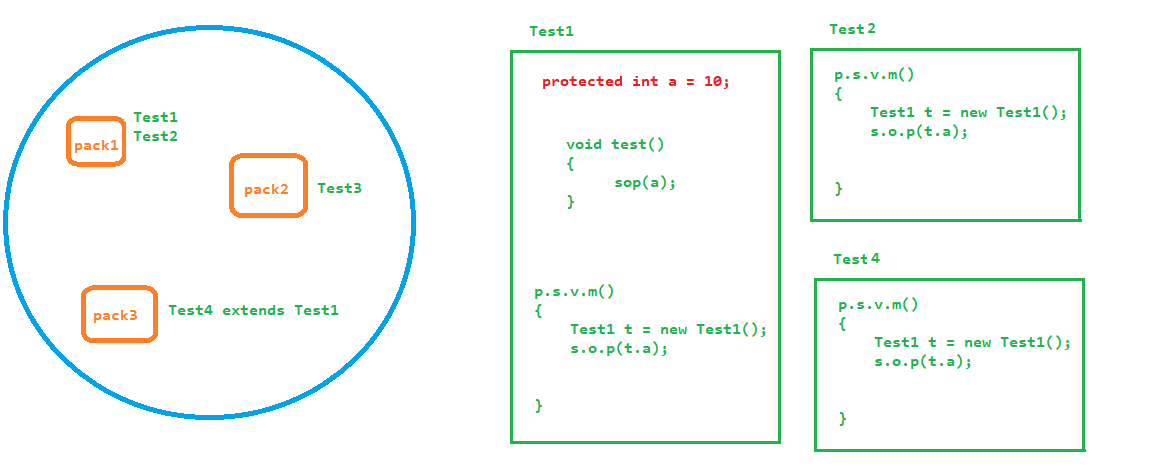
1. if it has any unimplemented methods / abstract methods

2. if you want to avoid creating an object to your class.

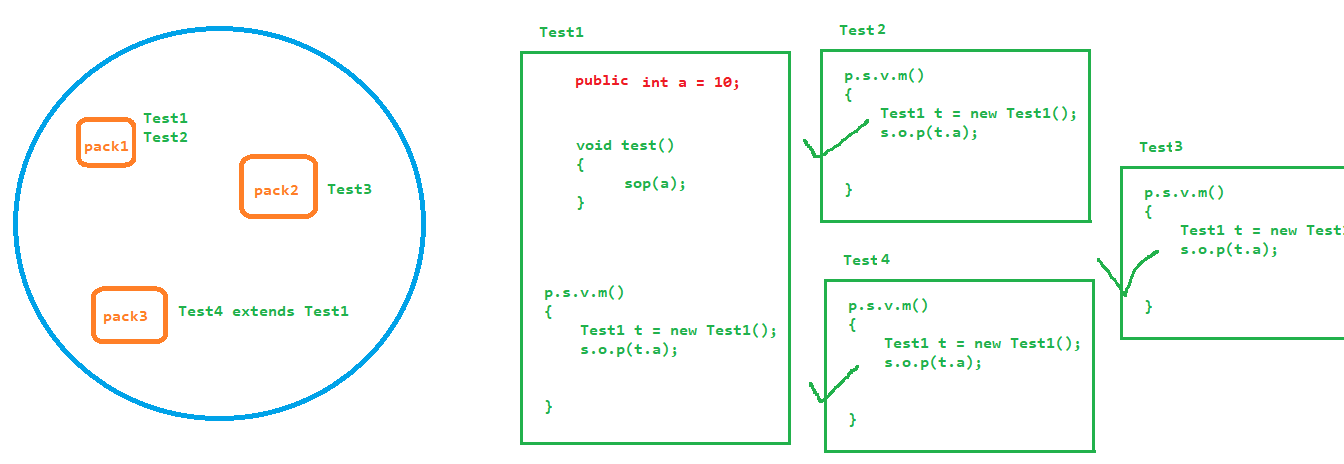
1. **Access Specifiers – *Specifies the accessibility of a member (variables, methods, constructors )***
   1. private
   2. package (default )

****

* 1. protected

****

* 1. public

****

1. **Access Modifiers – *Change the behaviour of a member (variables, methods, constructors )***
   1. static
   2. final
   3. abstract
   4. synchronized
2. **Interface**
3. **Enumerator**
4. **Strings**
5. s