**Software Engineer/software Developer**

**Developers :**

☞They can develop applications like whatsapp,paytm,googlePay,faceBook,twiter,banking App, telecom apps,Mobile apps.

**Software Tester**

☞Tester can test the application developed by developer

**✤Different types of application**

1. **Stand Alone application:**

⟹Any application runs on system its called as stand Alone /desktop/package application

1. **Web application**

⟹Any application runs on internet its called as web application

Eg. facebook gmail,bank application……….

1. **Distributed applications**

⟹One application interacting with another application is called as distributed application

1. **Mobile application**

⟹Any application runs on mobile application is called mobile application

**Vender:** sun MicroSystems takeover by OracleCorporation Pvt Ltd

**Author:** JamesGosling

**Version**: java16

**✤Why java?**

⟹Open source

⟹Independent to platform

**✤What is java?**

⟹Is a programing language

⟹Can develop applications like standalone /web/distributed applications……….

⟹Application can be run on any platforms

⟹Object oriented programming language

**✤Statement⟹** meaning full information

**✤Program**:⟹ set of statement

**✤Software**⟹ set of program

**✤API(application programming Interface):**⟹set of packages

**✤Packages:** ⟹set of classes and interfaces

**standAloneApplication: JSE API** —--> coreJava,jdbc

**web/distributed** : JEE API —--> jdbc,xml,jaxb,Restful Service

: JME API

*Note:* using J2SE API , we can develop standalone applications.

**✤Environment Set up**

☞J2sdk s/w (software)

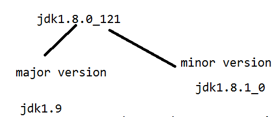
☞Eclipse IDE( **integrated development environment)**

☞Notepad ++

Once download jdk (java development kit) s/w install

Once installation is complete….. Two folders it will be generated

1. Jdk version
2. Jre version



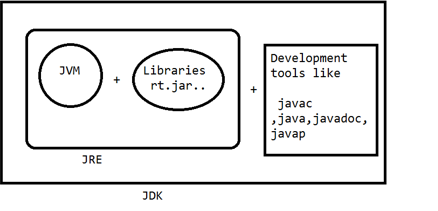
*Note:* If the micro version is changed then assume that previous version bug fixed up and released.

*Note:* Once changes happen on a major version assume that new features are added and released.

**✤JDK(java development kit)**  Developer +jdk

☞It provides the environment to develop applications or maintenance of the application

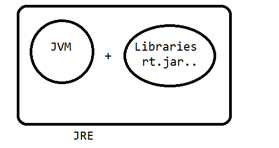
☞It represent JRE and development tools



**JRE( Java RunTime Environment)**  (client)

☞It provides the environment to maintain or run the application

☞It represents jvm(java virtual machine) and libraries(rt.jar)



**JVM(Java Virtual Machine)**

☞JVM provides the environment to compile and run the applications.

☞Jvm loads the application and verify and execute

**✤What is class?**

**⟹**class is a template or blueprint.

**⟹**Using class keywords we can create the structure of a java.

**⟹**Class can have variables and methods

**Syx**: class <logicalname/classname>

{

// methods

// variables

}

**Eg**.

Class welcome

{

}

**✤Method**

**⟹**Method represent set of statement to perform specific task

Use: Reusability

**Syx:** returnType logicalName/methodName(){ }

**✤What is the void method?**

**⟹**Void is a keyword , it doesn't return any information from the method

**✤Method can be classified two types**

**#userDefined methods**

**#Developer design methods**

Void add () void sub () void mul()

{

}

**✤Predefined methods**

**⟹**sunmicroSystem introduced predefined methods as part of the api

Public static void main(String[]args) —--> Execution will be started

{

}

**✤To print the information on console we can use**

system.out,println(“”);

**✤Compile java application**

Javac filename.java—> javac welcome.java

Public class welcome{

Public static void main(String[]args) {

system.out,println(“welcome to takeo”);

}

}

**✤.class contains bytecode format : 0 ,1**

**✤Compile time execution** :

Compile time it checks that any syntax mistakes are there or not…..

**✤Runtime exception:**

Application execution time it checks any logic mistakes then it generate Runtime exception

**✤What is variable:**

Variable can hold a value based on DataType

*Syx*: DataType variableName = variableValue;

**Data Types**

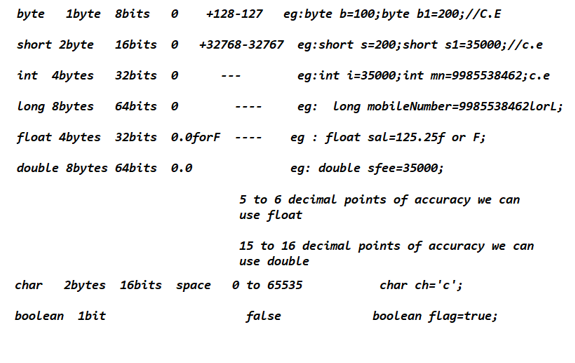
**✤DataType can classified types**

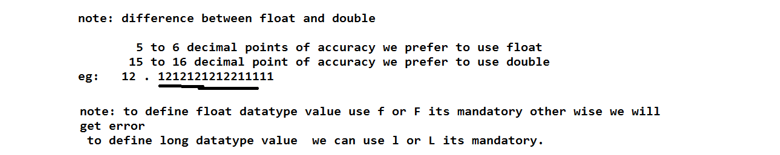
1. Primitive Data Type
2. Non-primitive data type
3. User Defined datatypes
4. Predefined data types

Using these primitive data types can be present numeric, logic,character,fraction,decimals

**1.Primitive dataTypes(8)**

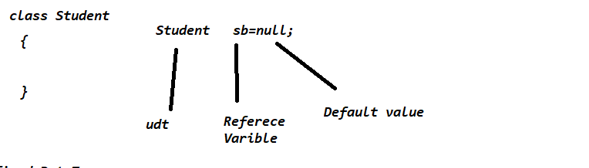
1. **Numerical DataTypes**
2. Integrated Data Types
3. Byte
4. Short
5. Int
6. Long
7. Floating point data types
8. Float
9. Double
10. **Character Data Types**
11. char
12. **Boolean Data Types**
13. Boolean





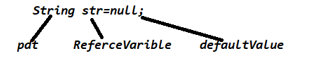
**✤Non-primitive data types**

1. **User Defined data types** :---->Using class name , we can define a data type is called a user defined data type

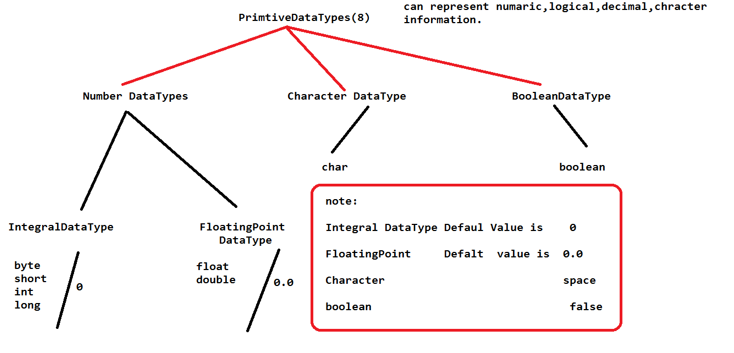


1. **Predefined data types**

Using a predefined class, we can define a datatype called a predefined data type. String is a predefined class, we can use as a dataType



**Chart**

****

**✤Compiler: —--->** Compiler.java to .class file

*Note:* compilation time it checks that syntactically having any mistakes or not if we have any mistakes it gives compiler error.

**✤Interpreter:** . **—---> .**class to execution

*Note:* Interpreter will read line by line statement and execute…. If we have any logical errors it generates RunTime Error.

Ctrl+shift+f = format

**Day4**

**Method:** set of statements to perform specific task

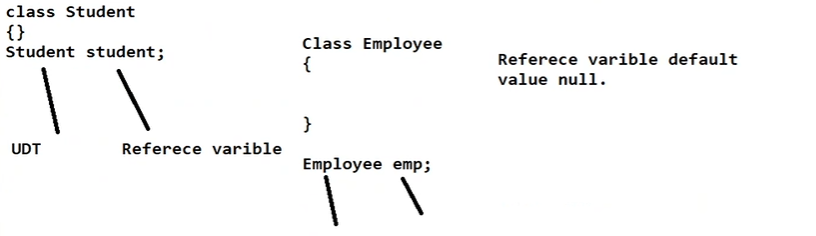
**Use:** Reusability

**DataTypes**:

**Primitive DataTypes :** byte,short,int,long,float,double,char,boolean.

**Non-primitive** dataType can classified two types:

1. **UserDefined DataType**: Using class name we can create data type



1. **Predefined dataType:**

Predefined class we can define a data type its called as predefined dataType

String str;

**Variables**

**Variable we can classify 3 types**

1. **Local variables**

* If we declare a variable inside the method it's called a local variable.
* JVM will not assign default value for local variables
* Scope of the local variable specific to method

Q. W.a.p. Swap two numbers with temp variables?

Q. w.a.p swap two numbers without temp variables?

Q. w.a.p swap three numbers with temp variables?

Q. w.a.p. Swap four numbers without temp variables?

**Declaring variable:**

DataType variableName;

Int fno;

Fno =100;

Declaration and initialization

DataType variablename =value;

**Identifiers:**

A name in a java program is called Identifiers, it may be classname,variablename,method name.

**Rules:**

1. a-z ,A-Z, 0-9,-,$ we can use as identifier

Abc123 —-->valid

\_$\_ —>valid

@\_$ —->not valid

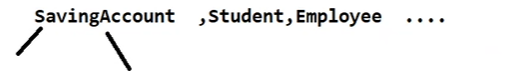
123abc —->not valid

1. Can not start with digit
2. Predefined keyword can not be used as a Identifiers
3. If,while,switch,break,return,.............53 key words
4. Predefined class names we can use as identifiers

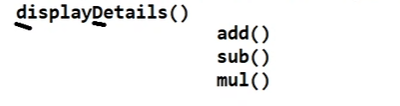
E.g String but don't write

Class String{}

1. className starts with Capital letters, if we have multiple words, first name starts with capital ,inner word starts with capital…



1. Method name start with small letter , if we have multiple words, first name start with small letters, inner word start with capital letter



1. Variable names start with small letters. If we have multiple words first name start with small letter, inner word start capital letter

Single line comment //

Multiple line comment /\* \*/

Documentation comment /\*\* \*/

**API (ApplicationInterface)**

**API** : Represents set of packages

**Package :** represents a set of classes and interfaces…

**Packages we can classified two types**

1. **Predefined packages**

Given by oracles

java.lang.\*;

java.io.\*;

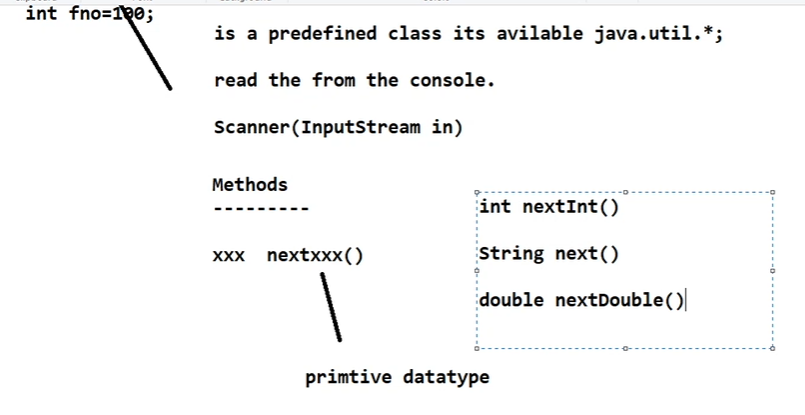
Every predefined class is associated with one of these packages,

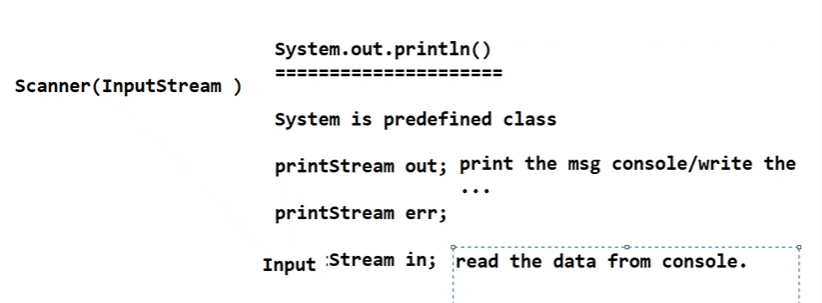
String is available

1. **userDefined packages**

Using package keyword can write the packages

Package packagename;





1. **Non-static variables/ instance variables**

* Non-static variables inside the class and outside the methods are called non-static variables.
* JVM will assign default values for non static variables.
* Scope of the non static variables specific to the object

**Rules:**

1. Non-static variables and Non-static methods can not Access Directly from Static methods.
2. If we access it generates C.E(compilation error)

How to access?

Using object reference variables can access Non-static variables and methods .

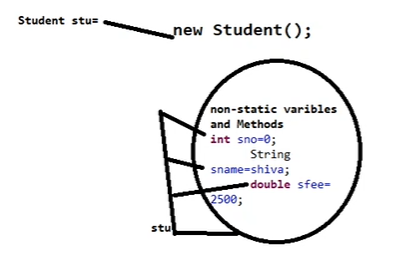
**What is an object? How to create an object?**

Object is an instance of a class, using a new keyword can create an object.

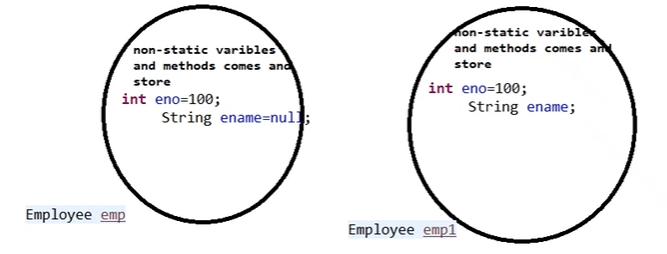
**Syx:** ClassName objRef= new ClassName();

Or UserDefined DataType

Using the class name we can define a datatype, it's called UDT.



2. Each and every object having separate memory



3. Reference variables once have null, can not access non-static variables and methods. If we are accessing it generating NullPointException.

.

1. **Static variables**

**Static:**

Static is a keyword, it acquires memory before execution of application.

Static variable /class variables:

If we declare inside the class, before the variable if we provide static keyword its called as static variable

Or before non-static variables if we specify a static keyword we can call it a static variable.

JVM will assign default values for static variables Scope of the static variable Entire class.

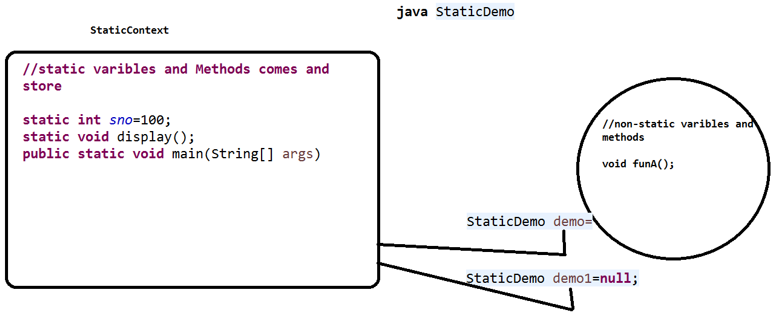
Variables which is declared static keyword we can call static variables

**Static method:**

Before the method if we use a static keyword we can call it a static method.

**Points to remember:**

1. Static variables and methods we can access Directly from static methods.
2. Static variables and methods we can access UsingClassName from static methods
3. Static variables and methods we can access UsingObject ReferenceVariables from static methods
4. Reference variable once we have a null,Even though we can access static variables and methods.
5. Static variables and methods we can access Directly from Non-static methods.



***Note:*** Reference variables once have an object or null by default it points to a static Context.

Because of this Reference variables once have a null we can access Static variables and methods.

**Note:**

**Local variables scope is specific to methods**

**Non-static variables scope is specific to object**

**Static variables scope is class.**

1) During Execution , any no of objects can be created for the same .class file

but the context of the class can be created only once for a given .class file

2) garbage collection of object will not have any effect on context of the class,

3) context of a class will be defined created before creating object of a class

but object of the class would be created only after the context of the class is created

**Day5**

**Operators in Java**

1. **Unary operators**
2. **Increment**

* ++expr : preIncrement
* Expr++ :postIncrement

1. **Decrement**

* - - expr :preDecrement
* Expr- - : post Decrement

1. **ArithmeticOperators**

+,-,/,\*,%

1. **ShiftOperator**
2. **Left shift operator**

<< is used to shift all of the bits in a value to the left side of a Right shift operator

1. **Right shift operator**

>> is used to move left operands value to right by the number of bits specified by the right operands

1. **Relational Operators**

>

<

>=

<=

==

!=

1. **Assignment Operators**

+= ,-= ,\*= ,/=,%=

1. **Ternary Operator**

(condition) ? st1:st2

**Note**: if condition is true st1 is executed otherwise st2 will be executed.

1. **Logical && and Bitwise &**
2. **Logical &&**

The logical && does not check the second condition if the first condition is false,It checks if the first one is true.

Both condition is true it returns true otherwise false

Int a =10;

Int b =5;

Int c =20;

system.out.println(a<b&&a<c);

**Example:**

if(uname.equals(“admin”) && pass.equals(“admin@123”));

1. **Bitwise &**

Bitwise & operator always checks both conditions whether the first condition is true or false .

Both conditions are true then only it returns true otherwise false.

system.out.println(a<b&a<c);

1. **Logical || and Bitwise |**
2. Logical || operator does not check if the first condition is true. It checks if the second condition is false. In these condition any one condition is true it returns true only
3. Bitwise | operator always checks both conditions whether the first condition is true or false . In these conditions any one condition is true it returns true only.

**Declaratie the variables**

Int a,b,c; —>valid

Int a,long b,c; —---> invalid

Int a,b,c = 300; valid

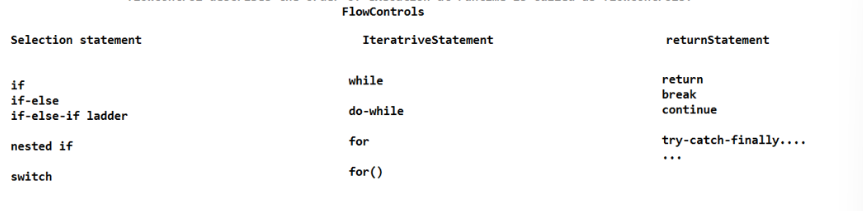
a= 0,b=0 c =300; valid

**Flow control**

Control the flow of execution at runtime its called as flow controls

Or

Flowcontrol describes the order of execution at runtime called flow controls.



**Selection Statement**

1. **If :**

‘If ‘checks the condition, if it is true the following statement will be executed, otherwise it will not execute .

If (condition) —-----> boolean expression

St1;

1. **If-else**

‘If’ checks the condition, if the condition is true the following statement will be executed, otherwise it will execute the ‘else’ block.

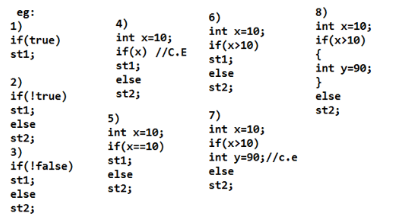
if(condition)

St1;

Else

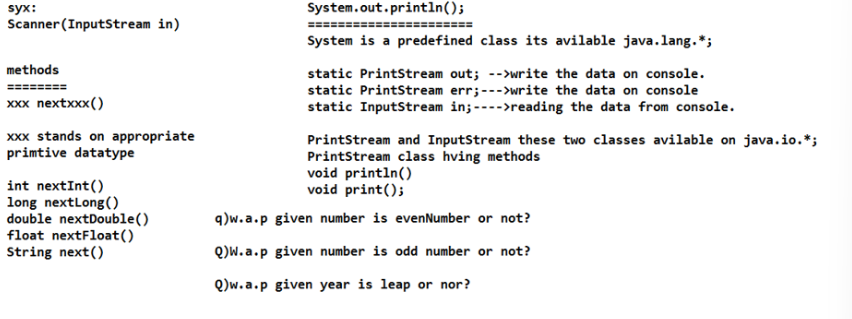
St2;

***Note***: ‘if’ follows at least one statement if we do not write it generates compilation error. More than one statement if we write we can represent using { } otherwise {} optional.



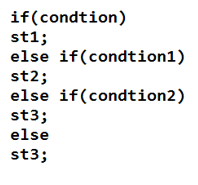
**Scanner:**

**Scanner**: Scanner is a predefined class using this we can read the data from console.This class available on java.util.\*;



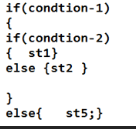
1. **If-else -if ladder**

This statement executes one condition from multiple statements.



1. **Nested if**

If inside one more if . if we write we can call it as nested if.

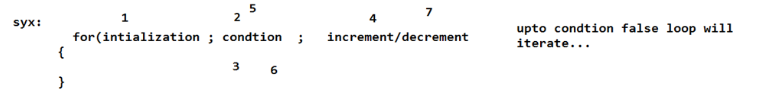


Iterate Statement

1. **for loop**

Using a for loop can iterate the statements one by one up to condition false.

Use: To perform fixed number of iterations ,we can use for loop



for ( int i =0 ; i<5; ++i)

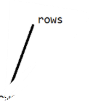
{

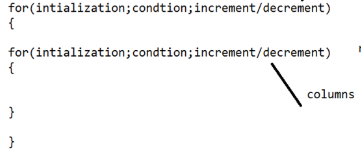
system.out.println(i);

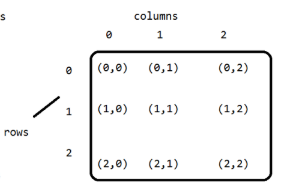
}

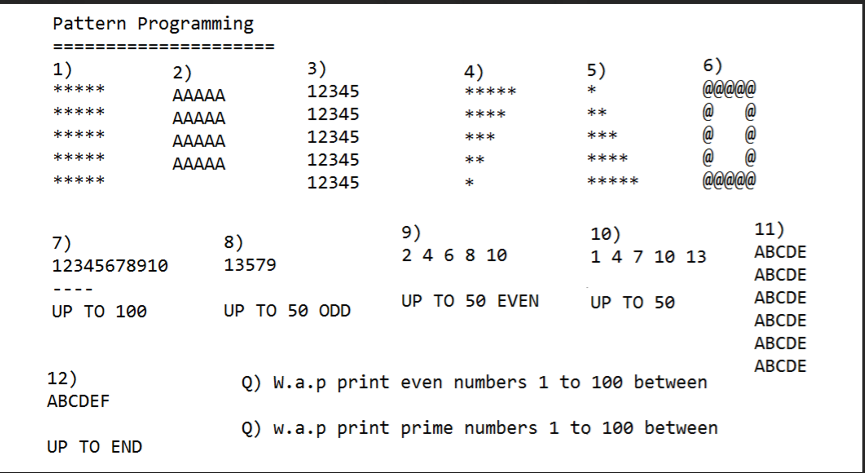
1. **Nested for loop**

‘for’ inside ,we can write one more for loop.



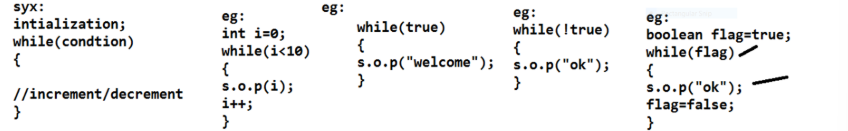






**While loop**

1. **While loop**: using while loop , we can iterate the elements one by one up to condition false. If the number of iterations is not fixed. We can use a while loop.



1. **Do while loop** : The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

### 

### **Syntax**

do {

*// code block to be executed*

}

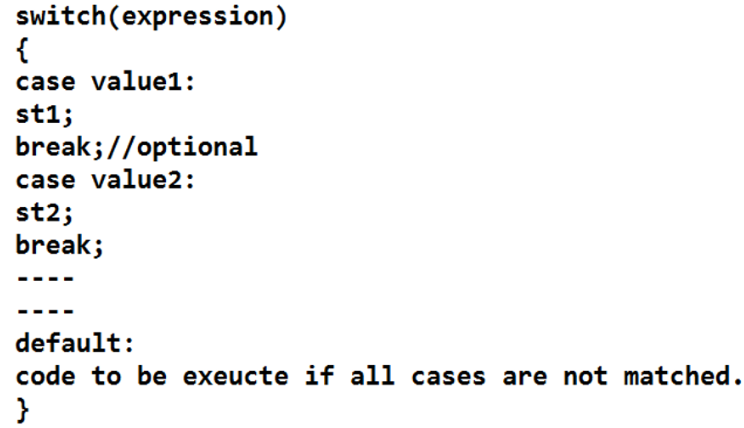
while (*condition*);

**### switch######**

**Switch**: switch executes one statement from multiple conditions. It accepts the arguments like byte,char, short,int,enum,String,and some wrapper classes .

**Points:**

1. Inside the switch, write at least one or N number of case values for a switch expression. And inside the switch , we can specify the case or default . it's mandatory, otherwise, we can print any statement.
2. The case value must be of switch expression type only.
3. Case value must be constant or literal . it doesn't allow variables.
4. Each case statement can have a break statement. Otherwise , once a case is matching following cases will be executed.
5. Duplicate case labels are not allowed.



**Assignments**

