COREJAVA-MODULE_1-NOTES

BY -AB

SYLLABUS

-INTRODUCTION OF JAVA

MODULE-1

1.BASIC STRUCTURE OF PROGRAM
2.ARCHITECTURE OF JAVA
3.DATA TYPES
4.VARIABLES
5.KEYWORDS AND IDENTIFIERS
6.OPERATORS AND TYPES
5.CONDITIONAL STATEMENTS
6.SWITCH CASE STATEMENTS
7.LOOPING STATEMENT
8.PATTERN PROGRAMMING
10.GENERAL PROGRAMS BASED ON FOR AND WHILE LOOPS

MODULE-2

11.METHODS AND METHOD OVERLOADING
12.VARIABLES AND ITS TYPES
13.ACCESSING DATAMEMBER IN MEMBERFUNCTION
14.EXECUTION PROCESS
15.CONSTRUCTORS
16.CONSTRUCTOR OVERLOADING
17.THIS KEYWORD
18.CONSTRUCTOR CHAINING
19.UML DIAGRAMS

MODULE-3

-OOPS

20.INHERITANCE
21.METHOD OVERRIDING AND SUPER KEYWORD
22.ABSTRACTION
23.INTERFACE
24.ACCESS SPECIFIER
25.ENCAPSULATION
26.POLYMORPHISM AND METHOD BINDING
27.GENERALISATION AND SPECILALISATION
28.TYPE CASTING
29.OBJECT CASTING

MODULE-4

30.OBJECT CLASS
31.STRING CLASS
32.ARRAY PROGRAMMING
33.EXCEPTION HANDLING
34.WRAPPER CLASSES
35.COLLECTIONS FRAMEWORK BY

36.MAPS
37.MULTITHREADING

Introduction

AUTHOR : JAMES GOSLING
VENDOR : SUM MICRO SYSTEM(ORACLE)
PROJECT NAME : GREEN TEAM
TYPE : OPEN SOURCE
INITIAL NAME : OAK
PRESENT NAME :JAVA
EXT :.java,.class,.jar
PRESENT VERSION : java14
OPERATING SYSTEM : ANY OPERATING SYSTEM
BASIS : C++
PRINCIPLE :WORA(write once run anywhere)

USES

-WebApplications
-MobileApplications
-ClientServerApplications
-EmbeddedSystems
-Robotics
-SAP

PARTS OF JAVA

1.J2SE/JSE(JAVA 2 STD EDITION)
2.J2EE/JEE(JAVA 2 ENTRPISE EDITION)
3.J2ME/JME(JAVA 2 MICRO EDITION)

Version History of JAVA:

JDK Alpha and Beta (1995)
JDK 1.0 (23rd Jan 1996)
JDK 1.1 (19th Feb 1997)
J2SE 1.2 (8th Dec 1998)
J2SE 1.3 (8th May 2000)
J2SE 1.4 (6th Feb 2002)
J2SE 5.0 (30th Sep 2004)
Java SE 6 (11th Dec 2006)
Java SE 7 (28th July 2011)
Java SE 8 (18th Mar 2014)
Java SE 9 (21st Sep 2017)
Java SE 10 (20th Mar 2018)
Java SE 11 (11 September, 25th 2018)
Java SE 12 (15 March, 19th 2019)
Java SE 13 (15 September, 17th 2019)
Java SE 14 (14 March, 17th 2020)
Java SE 15 (15Expected in September 2020)

FEATURES OF JAVA

A list of most important features of Java.

Simple:

-Simple to learn(Syntax's)
Object-Oriented :

Java is an object-oriented programming language. Everything in Java is an object.
Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and
maintenance by providing some rules.

Basic concepts of OOPs are:

Object
Class
Inheritance
Polymorphism
Abstraction
Encapsulation

Portable

-Can be used with any other language

Platform independent

Java code can be run on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc.
Java code is compiled by the compiler and converted into bytecode.
This bytecode is a platform-independent code because it can be run on multiple platforms,
i.e., Write Once and Run Anywhere(WORA).

-Secured

-any probleum happens only JVM will get effected but operating system is safe.
-Robust

-Most of things are automated
Ex:garbage collection
-Multithreaded

-Running multiple processeses(tasks) at same time.

Differences Between C and JAVA

C-lang	JAVA
1.C is a Procedural Programming Language.	-Java is an Object-Oriented Language
2.C was developed by Dennis M. Ritchie	-Java language was developed by James Gosling in 1995.
3.In the C declaration variable are declared beginning of the block.	-In Java, you can at the declare a variable anywhere.
5.Free is a variable used for freeing the memory in C	-A compiler will Call garbage collector for cleaning.
6.C does not support threading.	-Java has a feature of threading.
7.C support pointers.	-Java does not support pointers.
8.Memory allocation can be done by malloc. a	-Memory allocation can be done a new keyword.
9.Garbage collector needs to manage manually.	-it is automatically managed by a garbage collector
10.C does not have a feature of overloading functionality.	-Java supports method overloading.

MODULE-1

BASIC STRUCTURE OF PROGRAM

```
public class Demo
{
  public static void main(String args[])
  {
     System.out.println("This is my first program");
  }
}
```

Every program has 3- main parts

1.class declaration

Ex:public class Sample
   public class Demo

-It consists of 3 things

1.Access Modifier:It indicates that program is accessible to other user or not
     -There are 4 access modifiers in java public,private,protected and default
     -in above section class is public so it is freely accessible
     -All access modofiers will be in lower case.

2.class: class is a keyword(reserve word or predefine word) in java.
            -Every program must start with class keyword
            -all keyword must starts with smaller case so class-c is small

3.class Name:Every class has some name i.e class name or program name or file name
               -class name for standard should start with Capital letter
               -Java File name and class name must be same for remembering purpose
               -class name can only be combination of A-Z,a-z,0-9,$,_
Note:{//}---->scope of class

2.DEFNING MAIN METHOD

```
EX:  public static void main(String args[])
     {
           //LOGIC OF APP
     }
```
-it consists of 5 parts
1.AccessModifier
2.NonAccessModifier
3.return type

4.method name
5.command line arguments
1.Accessmodifier
---
public:
---
main() is public so that it is accessible to everyone freely.
Non Access Modifier

-we have 4 types of NAM--->static ,non static, final & abstract
-static---->method is accessible withoout object creation
-nonstatic--->method is accessible with object creation
Return type

-void---->it indicates that method is not going to return any value
method name

-if anyword contains()--->we can identified it as a method
Ex: main(),run(),display() etc
-main is name of method

command line arguments

String args[]
Theoretically--->we will called it as string arguments of array
String -s is capital(predefine class)
-This statements can be written in 3 ways
  1.String args[]
  2.String []args
  3.String[] args
Note:In synatx of main method only String-s is captial remaing all words starting letters are small.

3.Third part-Printing statement

System.out.println("This is my sample program");

-System :it is a pre define class

-out: it is an object (predefine)

-println():it is a method (predefine)

-In Simple println() is accesssed through out object but out object is present in system class.
-System is a classs which contains out object and out object is referring to println()
-Whever we gave in double qoutes that message will be printed as it is.

For Mobile Execution
---
please install this app from play store:

AIDE-IDE FOR C++ AND JAVA(ANDROID) or DECODER
JEDONA(IOS)

Some more sample programs

Ex1:

```
public class Info
{
  public static void main(String args[])
  {
     System.out.println("This is my first program");
     System.out.println("In my first class");
     System.out.println("Of learning core java");
  }
}
```

Ex-2:

```
public class AboutMe
{
  public static void main(String args[])
  {
     System.out.println("My Name is : Paul");
     System.out.println("I am 22 years old");
     System.out.println("I have graduated from Osmania University");
     System.out.println("My Ambition is to be a programmer");
  }
}
```

variations in printing message
---
println()--->print next message in new line.
print() --->print next message in same line

Example

```
public class AboutMe
{
  public static void main(String args[])
  {
     System.out.print("My Name is : Paul");
     System.out.println("I am 22 years old");
     System.out.print("I have graduated from Osmania University");
     System.out.println("My Ambition is to be a programmer");
  }
}
```

Commenting any line

-if we put // in starting of statement it will be commented(not considered as part of program)

```
  Ex: //public class AboutMe
```

-if we put // in ending of statement after that whatever we write it will not be considered as part
 of program.
   Ex: public class AboutMe//class declaration
-/*
    STMT1
    STMT2

    */
-Above comment section is used to provide description of programm in more than one line.
\n :-

```
public class MyInformation
       {
            public static void main(String args[])
            {
                System.out.println("Name : Qspiders  \n  DOB : 01-01-1992
\n Age:21 ");

            }
        }
```

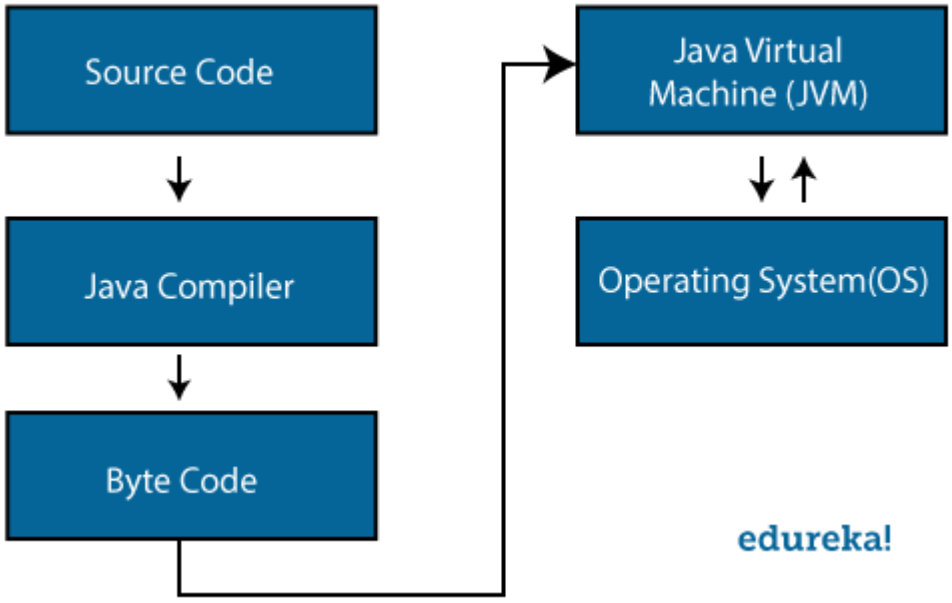
Notes: \n is used to break the line.
       -\n should must be in double quotes

Examples

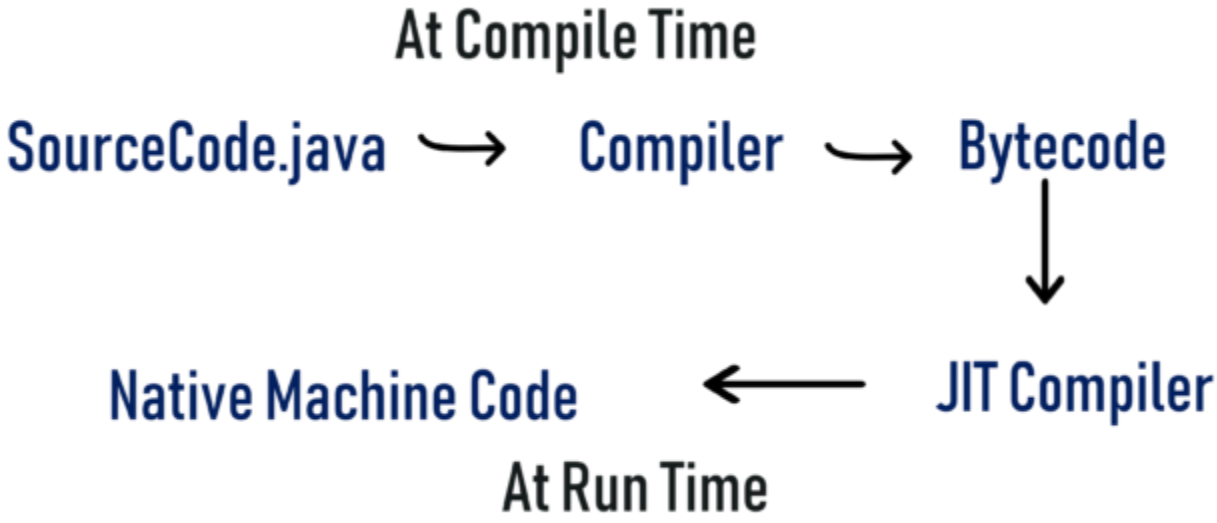
-ALSO WRITE COMMENT SETION FOR EACH LINE
1.WAP to print your information
   Output:
```
          Name : xyz
          Age  : xx
          DOB  : xx-xx-XXXX
```

2.WAP to print your information
   Output:
```
          Name: xyz
          YOP : xxxx
          Per :  xx%
          Str : xxx
          clg : abcd
```

ARCHITECTURE OF JAVA:-



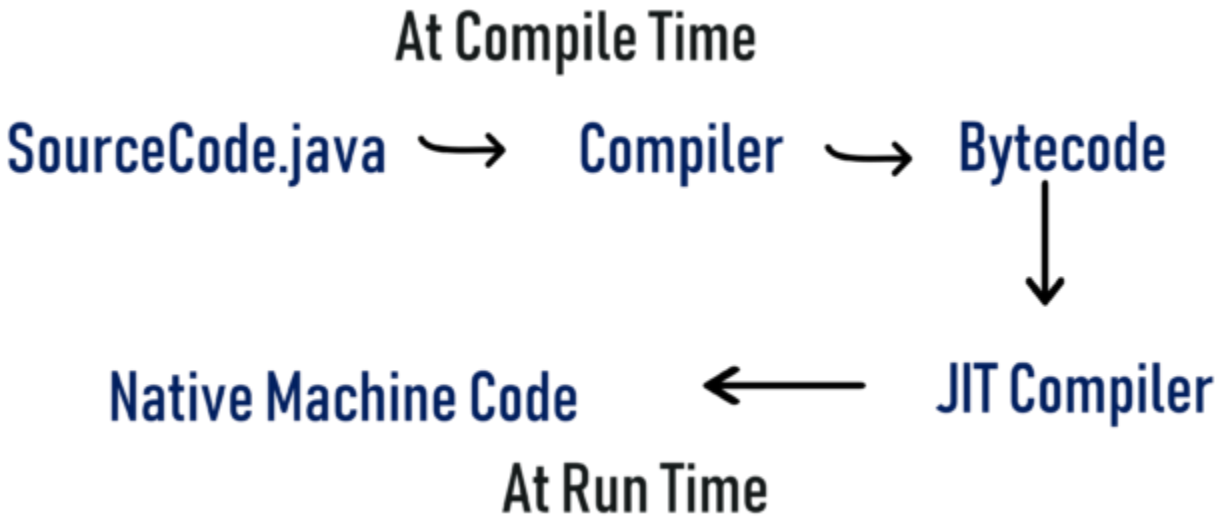

```
//REFER FIGURE OF ARCHITECTURE
Step-1:
--------
-Whatever program we write it is called as source code.
-source code should always save as ext ".java"
-Above prog should save as Hello.java

Step-2: Compilation
-------
-The process of converting our program into system understandable
form(byte code)
 is purpose of compiling a program
-To compile a prog go to cmd prompt and enter command as
  -javac progname.java
-Ex:javac Hello.java
-During compilation,compiler will check syntax errors like
[],;,(),{},:,spellings and case sensitivity.
-If any thing is wrong we will get compile time error.
-if nothing is wrong there is one class file get generated(byte code
file) with same name as .class.
```



step-3:Execution

-JVM-java virtual machine is responsible for execution of every java program
-it is like one software or one program.
-execution will happen in line by line manner
-during execution jvm will find logical error of program.
-for executiong program go to command prompt and enter command as
 - java programname
-Ex:java Hello
-Once we enter this commmand JVM will go to class file and take first line
and give to operating sys for execution,once OS responds that i understood that line

it sends second line and it continues till last line like this whole code of class file gets executed.

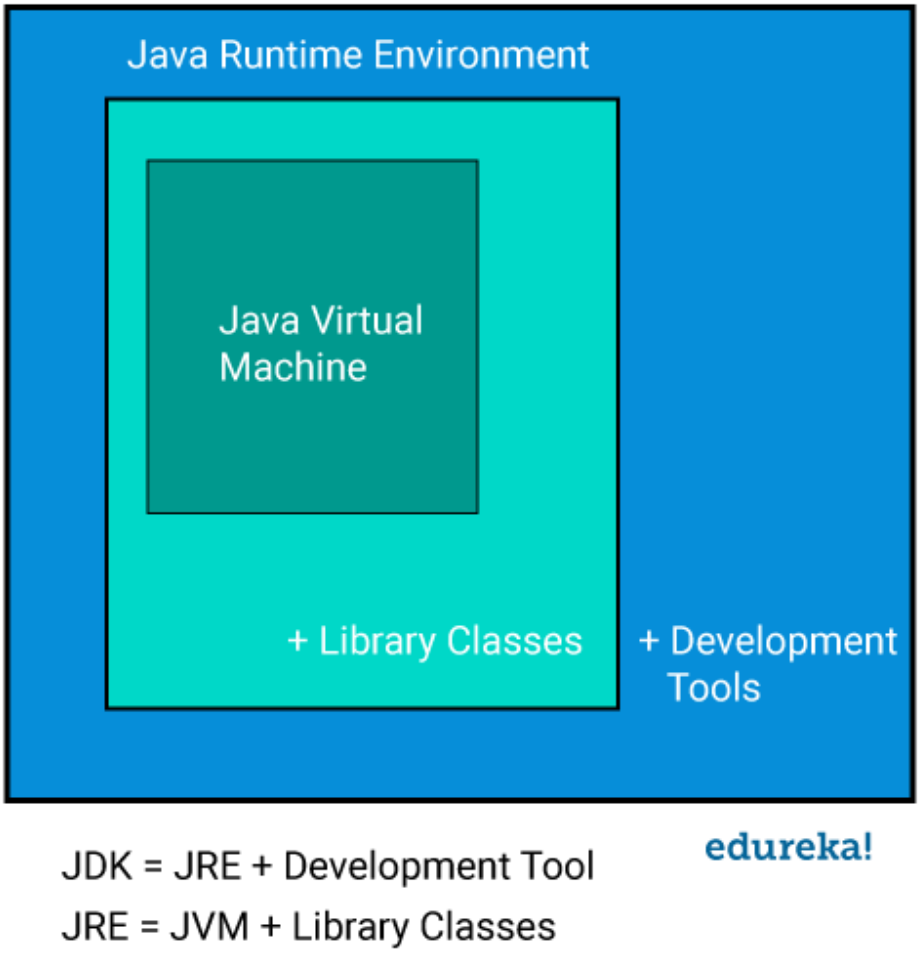


STEPS TO DESIGN A FIRST APPLICATION OF JAVA

STEP-1: Select n Editor(Notepad,NP++,Editplus or IDE-Integrated development tool)
STEP-2: Write the logic of APP
STEP-3: Save the APP
STEP-4: Compilation
STEP-5: Execution
Note:In IDE's 75% of work is automatic.
     Ex:Eclipse,NetBeans etc.

JDK-JAVA DEVELOPMENT KIT

-IF WE WANT TO DEVELOP AND EXECUTE JAVA PROGRAM IN OUR SYSTEM WE HAVE TO INSTALL JDK(JRE,JVM,SUPPORTING TOOLS AND SUPP CLASSES).



JDK: IT IS USED FOR DEVELOPING AND EXECUTING JAVA PROGRAM

JRE:JAVA RUNTIME ENVIRONMENT

-IT IS USED FOR EXECUTING JAVA PROGRAM OR RUNNING JAVA APP

JVM:JAVA VIRTUAL MACHINE

-IT IS RESPONSIBLE TO EXECUTE EVERY JAVA PROGRAM.

Note:when we install jdk with that JVM and JRE is available.

-How to install JDK?
-Refer the document ,which includes all steps

with screen shots you can see and install it. it is just 5 minute process.

Interview question

Q)can we change syntax of main method?
A)
-JVM is responsible to execute every java program.
- JVM
{
public static void main(command line args)
}
-Execution of every java program will always start from main method
-Because JVM only knows main method
-If main method is not there or main method syntax is changed our program
will not be executed.
-So, the ans is No we cannot change syntax of main method.

2.Can we keep main method as private?
A.No we cannot keep main method as private because private fields cannot be accessess from outside of class.
-And JVM needs to access main method from outside of class.
-If we keep main method as private program will compile but it will not be executed.

3.Can we keep main method as non static?
A.No we can't write main method as non static because if method is non static
we have to create an object and JVM can't create object on its own to access main method.

-If we keep main method as non static program will compile but it will not be executed.

Guidelines for executing program in command prompt

-Install JDK by referring the document send in whatSapp group

-If any problem with path setting (Refer Docs)

-Once it installed go to command prompt and type
java -version

-if java is successfully installed it gives the current version name which you installed

-if java is not installed properly it says "system cannot find cmd/path"

-Before writing program follow below step

-Create a folder at any particular location.
  For Ex: Go to C drive
           -Create a folder like "Java Programs"

-OPen note pad type ur program and save it with "class.java" Extension
  for ex:Sample.java

-For compilation go to command prompt -defaultly it will be like this- C:\Users\User>
-Meaning is we are C-drive Users-mainfolder User-sub folder

-For coming out of folder enter command as cd..
C:\Users>
-For coming out of folder enter command as cd..
C:>
-For entering into particular folder enter cmd as cd foldername.
-Since our program is in Java Programs we should enter cmdas
-cd Java Programs
C:\Java Programs>------>meaning we are in C-drive Java Programs - Mainfolder

-Compile the program as javac Programname.java
                      Ex:javac Sample.java
-Execute the program as java Programname
                      Ex:java Sample

-For clearing screen enter command as "cls"
-For changing drive enter cmd as drive name:
                                Ex: E:

DATA TYPES AND VARIABLES

1.Data
2.Data Types
3.Variables

Data:Any information is called as data.
For Ex:name,age,height,marks,percentage and salary etc

Data Type: it defines type of data.

-Divided into 2 types

1.primitive(System define)
2.Non primitive(user define)

primitive data type:

-These are system define data types

-These are fixed in there memory size
-these are 8 in numbers.

Name	Size	Examples	Default Values
1.byte	1 byte	10,2,5(127 is max)	0
2.boolean	1 byte or no size	true or false	false
3.short	2 byte	100,220....(32,768)	0
4.char	2 byte	A,a.....	emptyspace
5.int	4 byte	1,2,777......(2,147,483,647 is max)	0
6.float	4 byte	0.2,0.3,33.666.....	0.0
7.long	8 byte	33333333,6565655.....	0
8.double	8 byte	0.343434343,99.5555555.....	0.0

Note:
-When we want 4-6 digits of accuracy we go for float else we use double
-If we give more than 32 bit value as long it is always recommended to give l
 Ex: long contact =98786745341;

Non primitive DT: are not fixed in there memory size.

String : it is used to represents group of char's
ex:java,manual testing.....
Arrays:
{10,20...100}

Variables:

-Variables are used to store the data for printing or using it in future.
1.Variable Declaration

-Syntax for dec a variable is
  AccessModifier Datatype variablename;
   Ex: public int a;
       public float b;
       public char ch;
       public String s;
-variable name can be a combination of a-z,A-Z,0-9,$ and _.
-Whenever we declare a variable one memory block will get created

2.Variable Initialisation

Syntax: Variablename=value;

```
        a=100;
        b=0.3333f;//mandatory to write f
        ch='A';//mandatory to give ''
        s="java";//mandatory to give ""
```

-WE CAN DECLARE AND INITIALISE A VARIABLE IN SINGLE STATEMENT ALSO.
 syntax: Access Modifier Datatype varname=value;
          public int a=22;

Examples

1.int a=22,b=33;//valid
2.int a=33,b;//valid
3.float percentage=60.0;//invalid- f is missing
4.char ch='AB';//invalid-character can't be more than one char
5.float h=100f;//valid--100.0
6.int i=0.334;//invalid--integer can't store decimal values
7.String s="123";//valid--->System.out.println("123");
8.String d="3334+ghijk"//valid
9.double marks=100.3434d//valid---->in double d is optional
10.long number=939393939391//valid---> in long l is optional

How to print data stored in variable

Ex:int age=45;
   System.out.println(age);//45
   char grade='A';
   System.out.println(grade);//A

int-------->Data type
age-------->variable name
=---------->Operator
45---------->data/value
;---------->Termination of stmt

Excercise programs

1.WAP to print your name,age,height,last char of ur name using datatype
  and variables.

2.WAP to print your name,college name,YOP,Stream,marks and percentage
using
  data types and variables

3.WAP to print your name,aadhar number,email address and PANno using data
types
  and variables.

4.WAP to print your Accountnumber ,Name as per bank record,current
balance,type of account and
  min balance to be stored using data types and variables.

OPERATORS :-

1.Arithmetic Operators

+------>Addition
- ------>Subtraction
\------->division(output:Coefficient)
*-------->Multiplication
%-------->mode.(output :Remainder)

Example

```
public class Aops
{
      public static void main(String args[])
      {
       int a=10,b=2,c,d,e,f,g;
           c=a+b;
       System.out.println(c);
           d=a-b;
       System.out.println(d);
           e=a*b;
       System.out.println(e);
           f=a%b;
       System.out.println(f);
  }}
//Or we can also write as//
public class Aops1
{
public static void main(String args[])
{
System.out.println("One more way is");
System.out.println(a+b);
System.out.println(a-b);
System.out.println(a*b);
System.out.println(a/b);
}}
```

2.OPerator Overloading :Overloading is one thing but plays multiple roles.
---------- EX:A person can be son,father,husband,grandfather,friend,boyfriend etc

-Java doesnot support operator overloading but there is one exception case like + operator,
+ is an overloaded operator because it act as addition as well as concatination.

Addition

```
1. int +int(int/float/double/long/char/short/byte)
2. char+ char(int/float/double/long/char/short/byte)
    //Java provides unicodes for every character
       A-65,B-66,C-67.....................Z=90
       a-97,b=98,c=99.....................z=122
    Ex:'A'+65---->65+65--->130
        'a'+'Z'----->97+90-->187
    Ex: char ch=65;
        System.out.println(ch);//A
```

Concatination

-If any one operand is String plus operator will always act as concatination.

```
   Ex: String s="java";int a=123;
       System.out.println(s+a);//java123
       System.out.println("I am "+s+"  developer");
```

-After concatination result will always be string.

```
      System.out.println(123+" ");//"123"
```

3.Assignment operator (=)

-it is use to assign or store the vale intoa variable.

```
    Ex:int a=10,
       b=30;
       int c=a+b;
```

4.Comparison operator-(==)---->it compares values and gives output as boolean value

---

```
   int a=10,b=20,c=30,d=30;
System.out.println(a==b);//10==20//false
System.out.println(c==d);//30==30//true
     char ch='A',ch1='A';
System.out.println(ch1==ch);//A==A//true.
```

5.Relational Operators:checks relationship between two values and result will be boolean

---

```
>--------->greater
<---------->lesser
>=---------->greater than equals
<=----------->lesser than equals
!=----------->not equals
```

6.Logical Operator

And :

-it compares two input and if both inputs are true then output is true or else
false.

-It is represented as && (in below ex 0 indicate false,1 indicate true)

a	b	a&&b
0	0	0
0	1	0
1	0	0
1	1	1

OR :

-it compares two input and if any one inputs is true then output is true or else
false.

-It is represented as || (in below ex 0 indicate false,1 indicate true)

a	b	a||b
0	0	0
0	1	1
1	0	1
1	1	1

NOT

input	output
0	1
1	0

7.Unary Operators

1. Increment operator:It increases the value by one.for ex:a=10; increment a add +1 to a
   - -It is denoted as ++
   - -It is of two types
     1.pre-increment
     2.post-increment

2.Decrement operator:It decreases the value by one
   - -It is denoted as --
   - -It is of two types
     1.pre-decrement
     2.post-decrement

Types:

preIncrement:

-The rule is first increment value then print or assign it or store it in variable.
-it is denoted as ++variablename

```
For Ex:  int a=10;
         int b=++a;//pre increment
    Ex2: int a=250;
         int b=++a;//a is increaased by 1 and then 251 is stored in b
    Ex3: int a=50;//51
         ++a; //50+1
         System.out.println(a);//51
```

postIncrement:

-The rule is first print or assign it or store it in variable then increment value .
-it is denoted as variablename++

```
For Ex:  int a=10;//11
         int b=a++;//post increment
         System.out.println(b);//10
```

```
predecrement:
--------------
-The rule is first decrement value then print or assign it or store it in
variable.
-it is denoted as --variablename
  For Ex:  int a=10;
           int b=--a;//pre increment
      Ex2: int a=250;
           int b=--a;//a is decreaased by 1 and then 249 is stored in b
      Ex3: int a=50;//49
          --a; //50-1
           System.out.println(a);//49

postdecrement:
--------------
-The rule is first  print or assign it or store it in variable then
decrement value .
-it is denoted as variablename--
  For Ex:  int a=10;//9
           int b=a--;//post decrement
           System.out.println(b);//10

Excersice:
----------
1.int a=100;//101//102

  System.out.println(++a +  a++);
                     |      |
                     101 +  101-->202

2.int a=50;//51//52//51//50

  System.out.println(a++ + ++a + a--  + --a);
                     |       |   |        |
                     50 +    52+ 52   +   50----->204

3.int a=22;//21//20//21//22

  System.out.println(--a - a-- - a++ -  ++a);
                       |   |     |        |
                      21 - 21  - 20  -   22---->-42.
7.Combinational Operators
--------------------------
1.+= (compound addition assignment operator)
   For Ex: a=a+b; can also be written as
           a+=b;//by using combinational or compound operator
2.-= (compound subtraction assignment operator)
3.*= (compound multiplication assignment operator)
4./= (compound division assignment operator)
5.%= (compound modulo assignment operator)
```

Keywords

-These are the reserve words or predefine words which have some reserve meaning.
-Various keywords in java are.

1.Accessible keywords
---
public,private ,protected,static,final and abstract and return.

2.conditional keywords
---
if,else,else if,switch,case,break,continue,goto and const ,default.

3.iterative Keywords
---
for,while,do while

4.class level
---
class,package,import,extends,implements
5.Exception level
---
try,catch,throw,throws,finally
6.Others
---
volatile,transient,synchronised,native etc.

Note :

1.In java there is no word called as default but meaning is there
Ex: class A------->here access modifier is default
2.There is no word as non static but meaning is there
Ex: public void run()---->here non access modifier is non static
3.*All keywords must starts with smaller case

Identifiers

-These are the names given by programmer as per convention.
Ex: class name,variable name,method name and package name

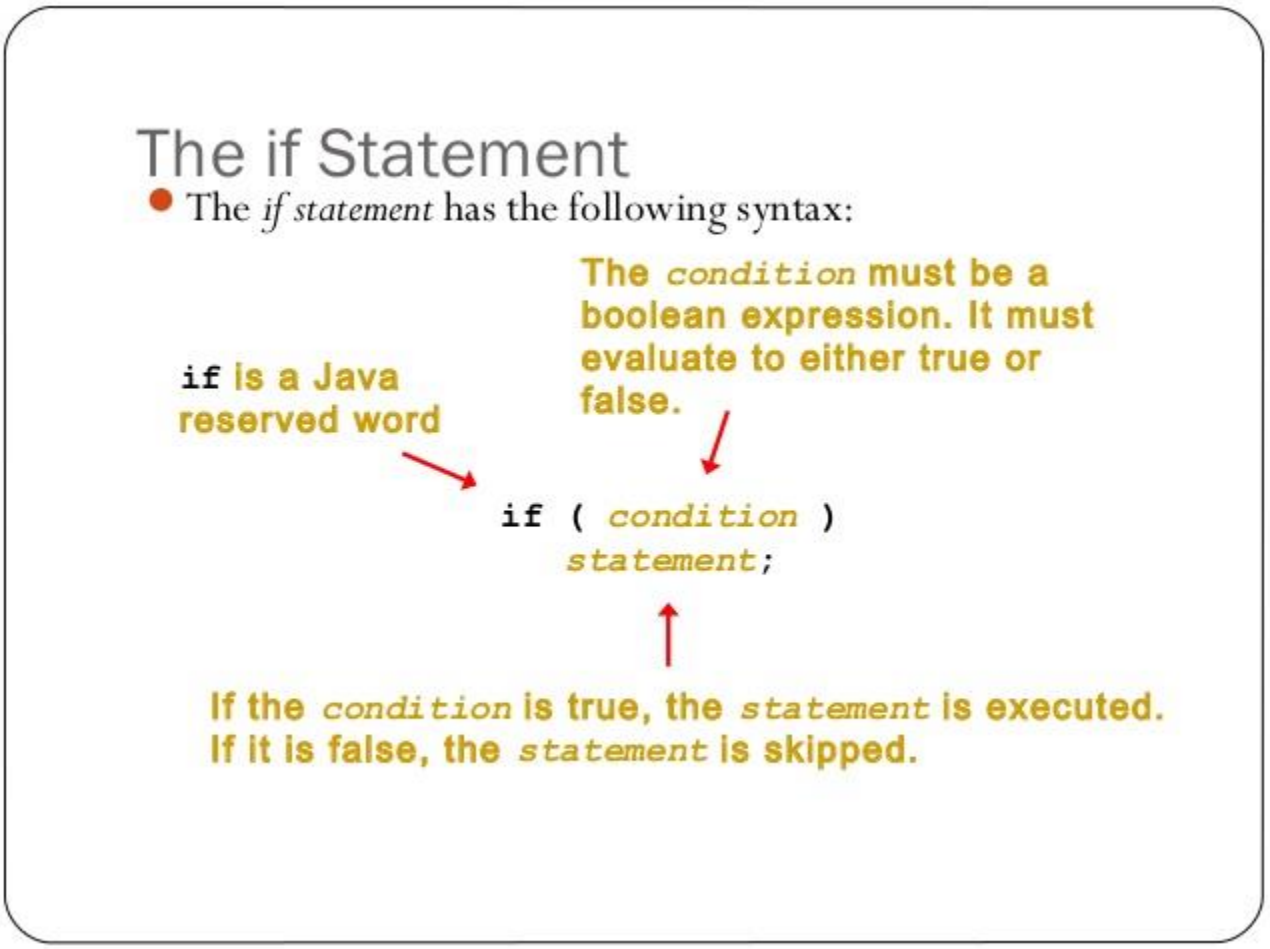
Rules for defining identifiers

1.An identifier can be a combination of A-Z,a-z,0-9,$ and _
but standard is:
class name : starts with capital
variable name: starts with small
method name: starts with small
package name: starts with small

2.If an identifier contains more than one word spaces are not allowed.
class My Program---->Invalid

```
   int   my age-------->invalid
   public static void display details()------->invalid

   class MyProgram//valid
   int mypercentage//valid

3.An identifier cannot starts with digit .
    class 1A---->invalid
    int 10a;------>invalid

     class A1---->valid
     int a10----->valid.

4.class name contains morethan one word for all words first letter should
be capital
    Ex:class MyFirsstProgram

5.If a methodname and variable name contains more than one word from
second word
  first letter should be capital
            Ex: int myAge;
            Ex: public static void displayDetails()


CONDITIONAL STATEMENTS
-----------------------
-Depending on conditions it switches control flow of execution from one
statement to another statement.

-Syntax
----------
type-1
------
if(Condition)//true means execute if part-------------------false means
execute else part
{
   //Set of statements//
}
else
{
  //set of stmts//
}
```

Note: writing else is not mandatory,we can add it as per our requirement.



Programs:-

1.WAP to find greatest of two numbers a=10 and b=20

```
 class Greatest
{
public static void main(String args[])
{
  int a=10,b=20;
  if(a>b)
 {
     System.out.println(a+" is an greatest number");
 }
 else
 {
    System.out.println(b+" is an greatest number");
}
}}
```

2.WAP to check whether the person is eligible to vote or not

```
class Voting
```

```
{
public static void main(String args[])
{
  int rajuage=23;
  if(rajuage>=18)
 {
     System.out.println(rajuage+"is eligible to vote as per
constitution");
 }
}}
```

3.WAP to check whether 22 is an even number or odd number.

```
class Even
{
public static void main(String args[])
{
  int a=23;
  if(a%2==0)//22%2(remainder)==0
 {
     System.out.println(a+" is an even number");
 }
 else
 {
    System.out.println(a+" is an odd number");
}
}}
```

4.WAP to check whether 56 is divisible by 5 or not.

```
class Even
{
public static void main(String args[])
{
  int a=56;
  if(a%5==0)//56%5(remainder)==0
 {
     System.out.println(a+" is divisible by 5");
 }
 else
 {
    System.out.println(a+" is not divisible by 5");
}
}}
```

5.WAP to check whether 10 is positive or negative

```
class NumbersP
{
public static void main(String args[])
{
  int a=-10;
  if(a>=0)
 {
     System.out.println(a+" is positive number");
 }
 else
```

```
 {
    System.out.println(a+" is negative number");
}
}}
```

Type-2

```
if(condition1)
{

}
else if(condition2)
{

}
else if(condition3)
{

}
else
{

}
```

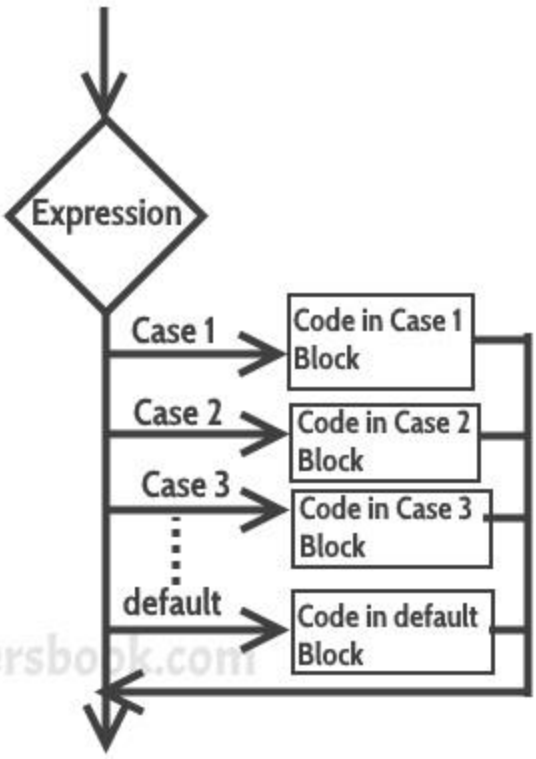
Type-3

```
if(condition1 LogicalOperator  condition2)
{


}
else
{

}
```

PROGRAMS

6.WAP TO FIND GREATEST OF 3 NUMBERS a=10 b=20 c=30
```
class NumbersP
{
public static void main(String args[])
{
  int a=10,b=20,c=30;
  if(a>b && a>c)
 {
     System.out.println(a+" is greatest numbers");
 }
 else if(b>a && b>c)
 {
    System.out.println(b+" is greatest of 3 numbers");
}
else
{
  System.out.println(c+" is greatest of 3 numbers");
```

```
}}}

7.WAP to check whether 22 is divisible by 3 and 5 or not
    int a=22
   if(a%3==0 && a%5==0)

8.WAP to check whether 55 is divible by 4 or 2
      int a=55;
    if(a%4==0 || a%2==0)

9.WAP to check whether a=30 and b=40 are equal or not
     int a=30,b=40;
     if(a==b)

10.WAP to check whether a=30 ,b=30  and c=40 are equal or not
        int a=30,b=30,c=40;
        if(a==b && b==c)

Switch case statements
-------------------------
syntax:
-------
switch(expression)
{
 case value1:
               break;

 case value2:
               break;

 case value3://
               break;
.....................so on

 default:
               break;
}
```



-We go for switch case statements if we have mutliplt test conditions to be checked
 in our program.
-Case values should be same type as Expression.

```
//WAP to take any number and print by refering below table
//0-zero
//1-one
//2-two
//3-three
//4-four
//5-five
//>5-invalid

public class Num
{
  public static void main(String args[])
 {
 int number=2;
 switch(number)
 {
 case 0:
System.out.println("Zero");
             break;//come out of switch case

 case 1:
System.out.println("One");
             break;//come out of switch case
```

```
 case 2:
System.out.println("Two");
            break;//come out of switch case
 case 3:
System.out.println("Three");
            break;//come out of switch case
 case 4:
System.out.println("Four");
            break;//come out of switch case
 case 5: System.out.println("Five");
            break;//come out of switch case
 default: System.out.println("Invalid");
            break;

}}}
```

break

-Break keyword is used to break the control flow .

-We can use breake keyword in switch case statements and in iterative statements also(Loops).

What if we Skip break keyword in case?
A.If we skip break keyword we will not get compile time error

-If there is no break keyword in case statement jVM will execute all the

cases irrespective of condition untill it finds next break statement

What if we skip break keyword in default?
A.it does not matter because after default nnothing is there to execute . so writing of break keyword is not mandatory in default statement.

Q>WAP to print daytype depends on daynum.

daynum	daytype
1,2,3,4,5	Weekdays
6	first weekend
7	Weekend
>7 or<1	Invalid

Hint :for case 1 to 5 do not write break keyword and give only one print statement

```
class Day
{
  public static void main(String args[])
 {
 int number=2;
 switch(number)
 {
 case 1:
 case 2:
 case 3:
 case 4:
 case 5: System.out.println("Weekday");
             break;//come out of switch case
 case 6: System.out.println("First Weekend");
             break;
  case 7: System.out.println("weekend");
             break;
}}}
```

Looping Statements
---
for loop

Looping statements

-Loop is define as repeated execution.

-If a part of code is repeateadly executing in our program rather than writing it
multiple times we can define it only once inside loop and run it as many times as we want

For example- print java 10 times or print 1 to 10 etc
-They are basically of 4 types

1.for

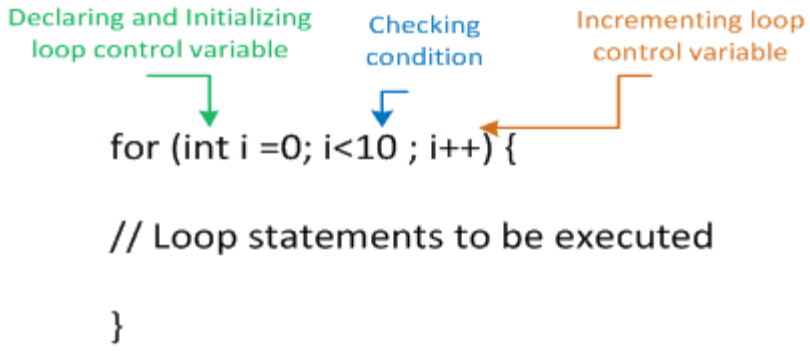
2.while

3.do while

4.for each(Enhanced or Advance for loop)

for LOOP

```
          for(initialisation;Condition;increment/Decrement)
           {
                 //Loop Body//
           }
```




```
Working
--------
1.step-1: initialisation
           Ex: int i=1;

2.Step-2: Checking Condition
          Ex:   i<10;
          if condition is true go inside loop and execute loop body till
last statement
          if Condition is false go outside loop

3.Step-3: increment or decrement the value as per requirement

4.repeate step-2  & step-3 until condition is  false.

Step-2 to Step-4 is called as iteration.

Programs
-----------
1.WAP to print java 10 times.
        //Without using loop//
class A
{
public static void main(String args[])
{
String s="java";
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
System.out.println("java");
}}
```

```
//USing loop
class A
        {
            public static void main(String args[])
            {
                for(int i=1;i<=10;i++)
                {
                    System.out.println("Java");
                }
}}
```

2.WAP to print the numbers from 1 to 100 using for loop

```
class B
        {
            public static void main(String args[])
            {
                for(int i=1;i<=100;i++)
                {
                    System.out.println(i);
                }
}}
```

3.WAP to print 100 to 1

```
            class C
             {
            public static void main(String args[])
            {
                for(int i=100;i>=1;i--)
                {
                    System.out.println(i);
             }}
```

Hint

- intialsiation is gretaer value; condition is > ; decrement
  int i=100; i>=1; i--
- initialisation is lesser value;condition is < ; increment
  int i=1; i<=100; i++.

4.

```
   class D
   {
       public static void main(String args[]){
        for(int i=10;i>=10;i++)
       {
          System.out.println(i);
       }}}
```

In Above program for loop will run infinite times because condition will never be false.

5.WAP to print a. 30 to 60
b. 65 to 20
c. A to Z
d. a to z

```
for(char ch='A';ch<='Z';ch++)
{
   System.out.println(ch);
}

for(char ch=65;ch<=90;ch++)
{
   System.out.println(ch);
}
```

6.WAP to print all even numbers from 1 to 50

```
for(int ch=1;ch<=50;ch++)
{
    if(ch%2==0)
   {
   System.out.println(ch);
}}
```

7.WAP to print sum of all even numbers from 1 to 20

```
  class S
 {
  public static void main(String args[]){
 int sum=0;
     for(int i=1;i<=10;i++)
     {
       if(i%2==0)
       {
         sum=sum+i;
     }}
      System.out.println(sum);
 }}
```

8.WAP to print product of all even numbers from 1 to 20

```
  class S
 {
  public static void main(String args[]){
 int pdt=1;
     for(int i=1;i<=10;i++)
     {
       if(i%2==0)
       {
         pdt=pdt*i;
     }}
      System.out.println(pdt);
 }}
```

9.WAP to print sum of first 10 natural numbers
    1+2+3..............+10

```
class Sum
 {
  public static void main(String args[]){
 int sum=0;
     for(int i=1;i<=10;i++)
     {
         sum=sum+i;
     }
      System.out.println(sum);
 }}
```

10.WAP to print Product of first 10 naturalnumbers
    1*2*3*..............*10

```
 class P
 {
  public static void main(String args[]){
 int pdt=1;
     for(int i=1;i<=10;i++)
     {
      pdt=pdt*i;
     }
      System.out.println(pdt);
 }}
```

11.WAP to print sum of all numbers disible by 5 present between 1 to 20

```
 class S
 {
  public static void main(String args[]){
 int sum=0;
     for(int i=1;i<=20;i++)
     {
       if(i%5==0)
       {
         sum=sum+i;
     }}
      System.out.println(sum);
 }}
```

12.WAP to print product of all numbers divisible by 5 present between 1 to 10

```
 class Pd
 {
  public static void main(String args[]){
 int pdt=1;
     for(int i=1;i<=10;i++)
     {
       if(i%5==0)
       {
         pdt=pdt*i;
     }}
      System.out.println(pdt);
 }}
```

13.WAP to count all numbers divisible by 5 present between 1 to 25

```
 class S
 {
  public static void main(String args[]){
 int count=0;
     for(int i=1;i<=25;i++)
     {
       if(i%5==0)
       {
         count=count+1;
     }}
      System.out.println("Final count:"+count);
 }}
```

14.WAP to print all numbers from 1 to 100 except multiples of 7

op: should not contains number which are divible by 7

```
class Div
{
  public static void main(String args[])
  {

      for(int i=1;i<=100;i++)
      {
        if(i%7!=0)
        {
         System.out.println(i);}}}}
```

15.WAP to print 5!

5*4*3*2*1

```
     class Factorial
     {
      public static void main(String args[]){
     int fact=1;
     for(int i=1;i<=5;i++)/for(int i=5;i>=1;i--)
       {
          fact=fact*i;
       }
     System.out.println(fact);
     }}
```

16.WAP to print multiplication table of 5 in table format

```
     5*1=5
     5*2=10
      .............5*10=50
     class M
     {
          public static void main(String args[])
          {
               for(int i=1;i<=10;i++){
               System.out.println("5 * "+i+"="+ 5*i);}}
```

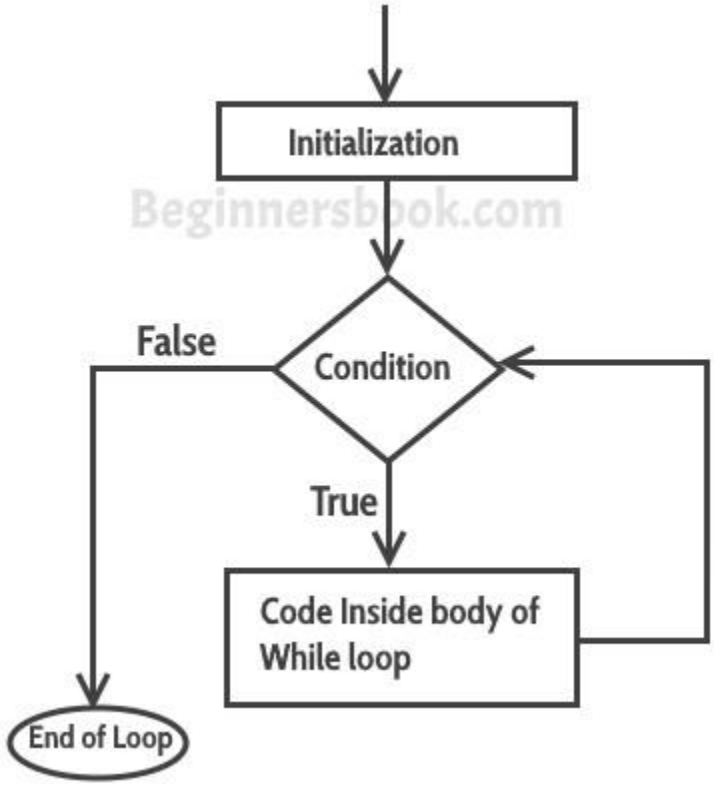
17.WAP to print fibnocci series
0 1 1 2 3 5 8 13

```
class Fibnocci
 {
       public static void main(String args[])
       {
           int f=0; f1=1,f2;
            System.out.print(f+" ");//0
            System.out.print(f1+" ");// 1///0 1
            for(int i=1;i<=6;i++)
            {
               f2=f+f1;//0 +1-->f2=1//1 +1//f2=2
               System.out.print(f2+" ");//0 1 1 2
               f=f1;//f=1//f=1
               f1=f2;//f1=1//f1=2
              }
}}
```

18.WAP to check Whether number is Prime Number or Not

```
public class Prime {

 public static void main(String[] args) {

     int num = 29;
     boolean flag = true;
     for(int i = 2; i <num; ++i)
     {
         // condition for nonprime number
         if(num % i == 0)
         {
             flag = false;
             break;
         }
     }

     if (flag==true)
         System.out.println(num + " is a prime number.");
     else
         System.out.println(num + " is not a prime number.");
 }
}
```

while loop



SYNTAX

```
 initialisation;

   while(Condition);
   {
      //some logic//
      increment/decrement
   }
```

```
ex: int i=1;//initialisation
     while(i<=5)//true//true.....//false
    {
        System.out.println("Java");
         i++;
    }
```

Programs

1.WAP to print numbers from 1 to 100

```
 class A
 {
   public static void main(String args[])
   {
      int i=1;
      while(i<=100)
      {
         System.out.println(i);
         i++;
       }
}
}
```

2.WAP to print numbers from 100 to 1

```
 class A
  {
    public static void main(String args[])
    {
       int i=100;
       while(i>=1)
       {
          System.out.println(i);
          i--;
        }
}
}
```

3.WAP to print A to Z

```
 class A
  {
    public static void main(String args[])
    {
       char ch='A'
       while(ch<='Z')
       {
          System.out.println(ch);
          ch++;
        }
}
}
```

4.WAP to print a to z

```
   class A
          {
            public static void main(String args[])
            {
               char ch='a'
               while(ch<='z')
               {
                  System.out.println(ch);
                  ch++;
                }
```

```
}
}
```

5.WAP to find reverse of a number

Algorithm:

step 1: take a num=123,rev=0.

step 2: take while loop and give condition as num>0

step 3: mode num with 10 and store it in rem variable

step 4:divide num with 10 and store it in num variable

step 5: rev*10+rem store it in rev

```
class Rev
 {
    public static void main(String args[])
    {
         int num=123,rev=0,rem;
         while(num>0)
          {
               rem=num%10;
               num=num/10;
               rev=rev*10+rem;
          }
         System.out.println(rev);
   }
    }
```

6.WAP to check whether number is Armstrong or not

Algorithm:

step 1: take a num=123,rev=0.

step 2: take while loop and give condition as num>0

step 3: mode num with 10 and store it in rem variable

step 4:divide num with 10 and store it in num variable

step 5: rev*10+rem store it in rev

```
class Arm
 {
    public static void main(String args[])
    {
         int num=153,rev=0,rem,temp=num;
         while(num>0)
          {
              rem=num%10;
              num=num/10;
              rev=rev+(rem*rem*rem);
          }
          if(temp==rev)
         System.out.println(temp+"is an Armstrong number");
         else
         System.out.println(temp+"is an NonArmstrong number");
      }
    }
```

7.WAP to check whether number is Palindrome or not

Algorithm:

step 1: take a num=123,rev=0.

step 2: take while loop and give condition as num>0

step 3: mode num with 10 and store it in rem variable

step 4:divide num with 10 and store it in num variable

step 5: rev*10+rem store it in rev

```
class Pal
 {
    public static void main(String args[])
    {
         int num=153,rev=0,rem,temp=num;
         while(num>0)
          {
              rem=num%10;
              num=num/10;
              rev=rev*10+rem;
          }
          if(temp==rev)
         System.out.println(temp+"is an Palindrome number");
         else
         System.out.println(temp+"is an NonPalindrome number");
      }
    }
```

do while

```
         do
         {
           //Some logic//
          }while(Condition);
class Demo
{
 public static void main(String args[])
 {
   int a=10;
   do
   {
     System.out.println(a);
     a++;
    }while(a<=13);
}}
```

Differences between loops

-for loop
---
we go for, for-loop if we know number of iterations priorly(already).

```
syntax: for(initialisation;condition;inc/dec)
        {

        }
```

while loop
---
we go for while loop if we don't know number of iterations.

```
synatx: initialisation;
        while(condition)
        {
         inc/dec;
        }
```

do while:
---
when we want our loop should run atleast one time then we go for do while loop

```
syntax:
         do
         {
            inc/dec;
           }while(condition);
```