Applet is deprecated...

BASIC METHODS IN APPLET

init() → initialis/ pass I/o.
start() → starts applet.
stop() → Stops applet.
Paint(Graphics g) → to draw something within applet.
destroy() → rem. apple from memory.

"NOT INCLUDED IN SYLLABUS"

ENCAPSULATION
#LECTURE - 07

* The object is core of JAVA, a class defines the shape & working of an object.
* class is a template/ framework of an object.
* A class can contain
  - → Fields → or "MEMBERS"
  - → Methods
  - → Constructors
  - → Blocks
  - → Nested classes & interfaces.

```
Circle class {
    double x,y,r;
    double circumference()
        return 2.314159*r;
    double area()
        return (22/7)*r*r;
}
```

```
class circledemo {
    Public static void main (string args[]) {
        circle c = new circle();
        c.x = 0.0;
        c.y = 0.0;
        c.r = 5.0;
        double p = c.circumference()
        double A = c.area();
    }
```

→ There should be a class which contains a single only main().
↳ IF NOT "COMPILATION ERROR"

→ The name of Java program file should be same as the name of main's class

→ We can pass any no. of arguments to a function.

CONSTRUCTOR:

The automatic initialization of var is done by CONSTRUCTOR.

It is a method.

It initializes an object immediately on creation.

This method has the same name as the class it resides in.

"WE CAN ALSO INITIALISE A METHOD"

It has NO RETURN TYPE.
Implicit return type is the class itself.
It initialise the "STATE OF THE OBJECT".

```
class circle {
    circle (double a, double b, double c) {
        x=a;
        y=b;
        r=c;
    }
}
```

@main⇒ circle c = new circle (1,0,2);

THIS KEYWORD:

it reduces "name-space collision"
Sometimes a method will need to refer to the OBJECT THAT INVOKES IT.

this - can be used inside any method
↓ to refer the CURRENT OBJECT

It is a keyword.

It is a REFERENCE TO THE OBJECT on which the method is INVOKED.

@ constructor.

```
Circle(      ) {
    this.x = a
    this.y = b
    this.r = c
}
```

CONSTRUCTOR OVERLOADING.

JAVA allows to declare one/more constr. method with DIFF LIST OF PARAMETERS & DIFF METHOD DEF.

```
circle (double x, double y) {
    this.x = x ; this.y = y
}
circle (double r) {
    x=0, y=0;
    this.r = r;
}
circle (circle c) {
    x = c.x;
    y = c.y
    z = c.z
}
circle() {
    x=0.0;
    y=0.0;
    r=0.1;
}
```

When multiple constructors exist
↓
this can be used from one constructor to invoke one of the other constr. of same class.

```
circle (double r){
    this (0.0, 0.0, r);
}
```

DEMONSTRATION - III
#LECTURE - 08

```
class A {
    A() {
        this(5);
        System.out.println("Hello, this is A");
    }
    A(int x) {
        System.out.println(x);
    }
}
```

O/P: 5
Hello, this is A.

```
class circle {
    double x, y, r;
    circle(double x, double y, double r) {
        this.x = x; this.y = y; this.r = r;
    }
    circle(double r) {
        this(0.0, 0.0, r);
    }
    circle(circle c) {
        this(c.x, c.y, c.r);
    }
```

JAVA PROGRAMMING INSIGHT
#LECTURE - 09.

Math.sqrt()
@ Java.lang.*

main():

It is a starting pt. of execution of main thread.
class that contains it is called main class
wkt, name of file is same as class main.

```
public static void main(String args[])
```

- public ↑ ACCESS MODIFIER
- static ↑ RETURN TYPE
- void ↑ Name method
- main ↑ String class
- (String args[]) ↑ array of String objects.

In a class if we create a object then for to access it method, we should have created a object in the first place.

BUT, if STATIC ⇒ then without creating any object of this class we can access its methods.

without obj u can call this METHOD.

args[] is array of any size.
array of string object.

Public ⇒ member may be accessed by code outside the class in which it is declared.

DEFAULT: Public.

Keyword static allows main() to be called without having to instantiate a particular instance of a class.

↳ this is necessary because JAVA INTERPRETER main is called by JI before any obj are made.

Java sees everything as string object.

System.out.println

System → final class Java.lang package.
out → class variable of type "PrintStream" declared in System class.

println → It is a method of PrintStream class.

+ for concatenation. print Vs println.
+"\n" NEXT LINE.

We can also use printf = formal. ~ that of C.

JAVA RUNTIME DATA INPUT

Command line input.

Hi this is prabha
args[0] [1] [2] [3].

Scanner class

Scanner → It is one of the predefined class which is used for reading the data dynamically from the keyboard

It is in utility (util) package. ie) import Javautil.Scanner.

```
Scanner s = new (System.in); # Constructor Scanner class object.
num = S.nextInt();

While ( input.hasNextInt()) {
        l.add(input.nextInt());
}
```

ctrl+Z Termination of scanning.

```
for:i→l.size()
l.get(i)        l.size()
```

DataInputStream

```
import java.io.*;
DataInputStream in = new DataInputStream(System.In);
System.out.flush();
temp = in.readLine();
Intrest = Float.ValueOf(temp);
Years = Integer.parseInt(temp);
```