Day 15

Instance variable:

- Inside class outside ~~variable~~ method
- parameter = instance variable store
- = operator
- instance var of parameter will be same. (it won't work)

pointer Refer to all

```
this.empid = empid;
this.salary = salary;
this.designation = designation;
```

this pointer:

- find diff b/w instance variable of parameter
- if instance (v) of parameter are same we will use this pointer

(3) Using Constructor:

- Special type of method
- name same as class name
- It will not return any value (not even void)
- Automatically called time of obj creation.
- overloading is possible (multiple constructor same name create)

Use of Constructor:

used for initializing the instance variables

2 Types:

(1) Default

(2) Parameterized

Default Constructor:

eg: public class Empconstructor {

```
int empid;
int salary;
String designation;

public Empconstructor()
{
```

Constructor →

// initialize instance var

```
empid = 100;
salary = 10000;
designation = "Developer";
```

I) public void display() → small name → Method

II) public Display() → class name → Constructor

```
}
```

PSVM. {

Automatically called when obj created

classname Emp constructor e = new Emp constructor();

Sysout (e.empid);

" (e.salary);

" (e.designation);

}

Parameterized Constructor:

~~this.empid = empid~~

public Defaut_constructor(int empid, int sala
String designation)

~~emp~~.

this.empid = empid;

this.salary = salary;

this.designation = designation;

}

public void disp()

{

empid

}.

PSVM {

= new (15, 5000, "Developer")

}.