Mahindra École Centrale
COLLEGE OF ENGINEERING

# Object Oriented Programming JAVA

Dr. Prafulla Kalapatapu
Computer Science Engineering
Mahindra Ecole Centrale
prafulla.kalapatapu@mechyd.ac.in

Mahindra École Centrale
COLLEGE OF ENGINEERING

# ENCAPSULATION

# Constructor

Mahindra École Centrale
COLLEGE OF ENGINEERING

- What is a constructor?
  It is a special method.

- What is the purpose of the constructor?
  To initialize object's state. (Assigning values to instance variables).

- Rules to write a constructor.
  - classname == methodname
  - It should not have return type

# Constructor Vs Method

Mahindra École Centrale
COLLEGE OF ENGINEERING

| Constructor | Method |
|---|---|
| It is a special method i.e it calls implicitly when object is going to be created as a 3rd step. | It is a method i.e it calls explicitly |
| classname == methodname | It can be any name |
| It should not have return type | It should have return type |
| Purpose is to intialize object state | Purpose is to perform some task |

# Case (i)

Mahindra École Centrale
COLLEGE OF ENGINEERING

- If we don't write a constructor, java compiler will write default constructor

Ex:

```
class Abc
{
}
```

- In the above class Abc, we didn't write any constructor, when you compile this class, your java compiler will write default constructor as shown below.

Ex:

```
class Abc
{
    Abc()
    {
        super();
    }
}
```

Written by the java compiler

It calls the super class constructor

Note: Any class it is implicitly derived from an Object class

# Case (ii)

Mahindra École Centrale
COLLEGE OF ENGINEERING

- If we write a constructor in the class

Ex:

```
class Abc
{
    int a;
    Abc()
    {
        super();
        a=20;
    }
}
```

→ Written by java compiler after compiling

NOTE: Default constructor is written by java compiler, only when there is no constructor in the class

- After compiling above class, your java compiler will add one statement as first line in the constructor

super();

- What is the responsibility of super();

  super() calls the super class's constructor.

# Example Programs

1. Create an Employee class with the following instance variables eid, ename, ebasic, eda, epf, egross, etax, enet, eded and write the methods with following names void calcDed(),void calNetSal() and void displayEmpDetails().

```
class Employee{
int eid,ebasic;
String ename;
double eda,epf,egross,etax,enet,eded;
void calcDed(){
eda=ebasic * 80/100.0;
egross=ebasic+eda;
epf=egross*12/100.0;
etax=egross*10/100.0;
eded=epf+etax;
}
void calNetSal(){
enet=egross-eded;
}
void displayEmpDetails(){
System.out.println(eid+"\nename="+ename+"\nnet="+enet+"\negross"+egross+"\nepf="+epf+" "+etax
+" "+eded);
}
public static void main(String []a)
{
Employee e1=new Employee();
e1.eid=100;
e1.ename="ABC";
e1.ebasic=6000;
e1.calcDed();
e1.calNetSal();
e1.displayEmpDetails();
}
}
```

2. Create an Student class with the following instance variables sid, sname, smarks, sbranch and sgrade and write the methods with following names void calcGrade() and void dispStdDetails().

```
class Student{
int sid,smarks;
String sname,sbranch;
char sgrade;
void calcGrade(){
if(smarks>=80)
sgrade='A';
else if((smarks>=60)&&(smarks<80))
sgrade='B';
else
sgrade='C';
}
void dispStdDetails(){
System.out.println(sid+"\nsname="+sname+"\nmarks="+smarks+"\nsbranch"+sbranch+"\nsgrade="+sgrade);  }
public static void main(String []a) {
Student s1=new Student();
s1.sid=123;
s1.sname="ABC";
s1.smarks=82;
s1.sbranch="csis";
Student s2=new Student();
s2.sid=132;
s2.sname="XYZ";
s2.smarks=65;
s2.sbranch="csis";
s1.calcGrade();
s1.dispStdDetails();
s2.calcGrade();
s2.dispStdDetails(); } }
```