**Java Programming**

**Example: Create personal details**

Public class personal

{

Private string name,gender;

Private int age;

public personal(string n,string g,int a)

{

name=n;

gender=g;

age=a;

}

public string getname()

{

return name;

}

/\*

public void setname(string name)

{

this.name=name;

}

\*/

public string getgender()

{

return gender;

}

/\*

public void setgender (string gender)

{

this. gender = gender;

}

\*/

public string getage ()

{

return age;

}

/\*

public void setage (int age)

{

this.age= age;

}

\*/

}

Public class Academic extends personal

{

Private string standard, university;

public Academic(string a, string u, string name, string g, int ag)

{

**super**(name,g,ag);

standard=a;

university=u;

}

public string getstandard ()

{

return standard;

}

/\*

public void setstandard (string standard)

{

this.standard =standard;

}

\*/

public string getuniversity ()

{

return university;

}

/\*

public void setuniversity (string university)

{

this.university = university;

}

\*/

In examples,

Academic is **sub-class or derived class**

Personal is **Super class or parent class**

* If any super class method is re-implemented or redefined in sub-class then it is called **over-raiding and use that method as ‘Abstract’.**
* **When we create multiple methods with same name then it is over-loading.**

**Database:**

Select Oracle database 10g-Run SQL command line

SQL- connect

Enter user name: scott

Enter password:

Connected.

SQL- create table- table name banksys(name varchar2(26), id number(26));

Now table is created. Then insert the values.

SQL- insert into banksys values(‘laksmi’,1002);

1 row created i.e one record is inserted.

If we want to add one more colum to existing colum called gender:

SQL- alter table banksys add gender varchar2(26);

Now table is altered.

SQL- desc banksys;

To remove the column gender:

SQL- alter table banksys drop column gender

Table is altered.

To c if it is removed then

SQL- desc banksys;

If we have to insert multiple records:

SQL- insert into banksys values(‘&name’ ,&id);

Enter value for name- rakesh

Enter value for id- 1006

Old 1: insert into banksys values(‘&name’ ,&id)

new 1: insert into banksys values(‘rakesh’ ,1006)

1 row created.

SQL- /

**‘/’** is used if we have to insert one more record.The same old 1 and new 1 lines are created.

**‘\*’** is used to display the values

If we have to display only name related values:

SQL – select name from banksys

We get names laksmni,rakesh etc.

SQL- select id from banksys

We get id’s 1002,1006 etc

If we have to delete one row:

SQL- delete from banksys where id=1002

1 row will be deleted.

If we have to delete 3 rows

SQL- delete from banksys;

3 rows deleted.

If we have to save the file, we type **‘commit’**.

**Database Connection:**

import java.sql.Connection;

import java.sql.Statement;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.PreparedStatement;

import java.util.Scanner;

import java.sql.ResultSet;

import java.sql.CallableStatement;

public class DatabaseConnection

{

Connection c;

Statement s;

PreparedStatement p;

String name,password,gender;

ResultSet r;

CallableStatement cs;

int number;

int count=1;

public void CreateConnection()

{

try

{

//Normal Statement

/\*