Q.N.1) Create a class of a “Human Being” . Define its possible properties and methods using appropriate access modifiers. Create a Derived child classes called “male” and “female” using the concept of inheritance.

a) Show how encapsulation concept works for this.

b) Show how child class can use parent class members.

c) Create objects of each class and access members.

Ans.

**public** **class** HumanBeing {

**private** String male\_name;

**private** String female\_name;

**void** setMaleName(String m) {

**this**.male\_name = m;

}

**void** setFemaleName(String f) {

**this**.female\_name = f;

}

**public** String getMaleName() {

**return** male\_name;

}

**public** String getFemaleName() {

**return** female\_name;

}

**void** run() {

System.***out***.println(male\_name +" is Running");

}

**void** eat() {

System.***out***.println(female\_name +" is Eating");

}

}

**class** Male **extends** HumanBeing{

**public** **static** **void** main(String[] args) {

Male pritam = **new** Male();

pritam.setMaleName("Pritam");

pritam.run();

}}

**class** Female **extends** HumanBeing{

**public** **static** **void** main(String[] args) {

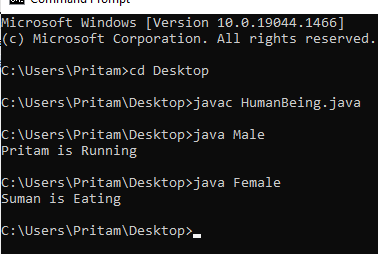
Female suman = **new** Female();

suman.setFemaleName("Suman");

suman.eat();

}}

**OUTPUT**



Q.N.2) Create a class of a “Player”. Create child classes called “Footballer” and “Cricketer”.  Create child classes of “Footballer” and call it “GoalKeeper”, ‘Midfielder“ and “Defender”. Create child classes of “Cricketer” and call it “Bowler”, “Batsman” and “allrounder” using the concept of inheritance. Define possible properties and methods of each class using appropriate access modifiers

Show how encapsulation concept works for this.

Write which type(s) of inheritance this solution would take.

Create objects of each class and access members.

Show how child classes can use parent class members.

Ans.

**public** **class** Player {

**private** String Name;

**void** football(String a) {

System.***out***.println(a + " Plays Football");

}

**void** cricket(String b) {

System.***out***.println(b + " Plays Cricket");

}

**void** setName(String name) {

**this**.Name = name;

}

String getName() {

**return** Name;

}

}

**class** Footballer **extends** Player{

**void** team(String c) {

System.***out***.println(c +" is a National Player");

}

}

**class** Goalkeeper **extends** Footballer{

**public** **static** **void** main(String[] args) {

Goalkeeper keshab = **new** Goalkeeper();

keshab.setName("Keshab");

keshab.football(keshab.getName());

System.***out***.println(keshab.getName() +" is a Goalkeeper");

keshab.team(keshab.getName());

}

}

**class** Midfielder **extends** Footballer{

**public** **static** **void** main(String[] args) {

Goalkeeper manoj = **new** Goalkeeper();

manoj.setName("Manoj");

manoj.football(manoj.getName());

System.***out***.println(manoj.getName() +" is a Midfielder");

manoj.team(manoj.getName());

}

}

**class** Defender **extends** Footballer{

**public** **static** **void** main(String[] args) {

Goalkeeper sachin = **new** Goalkeeper();

sachin.setName("Sachin");

sachin.football(sachin.getName());

System.***out***.println(sachin.getName() +" is a Defender");

sachin.team(sachin.getName());

}

}

**class** Cricketer **extends** Player{

**void** team(String d) {

System.***out***.println(d +" is a State Player");

}

}

**class** Bowler **extends** Cricketer{

**public** **static** **void** main(String[] args) {

Cricketer pritam = **new** Cricketer();

pritam.setName("Pritam");

pritam.football(pritam.getName());

System.***out***.println(pritam.getName() +" is a Bowler");

pritam.team(pritam.getName());

}

}

**class** Batsman **extends** Cricketer{

**public** **static** **void** main(String[] args) {

Cricketer barun = **new** Cricketer();

barun.setName("barun");

barun.football(barun.getName());

System.***out***.println(barun.getName() +" is a Batsman");

barun.team(barun.getName());

}

}

**class** Allrounder **extends** Cricketer{

**public** **static** **void** main(String[] args) {

Cricketer sanskar = **new** Cricketer();

sanskar.setName("Sanskar");

sanskar.football(sanskar.getName());

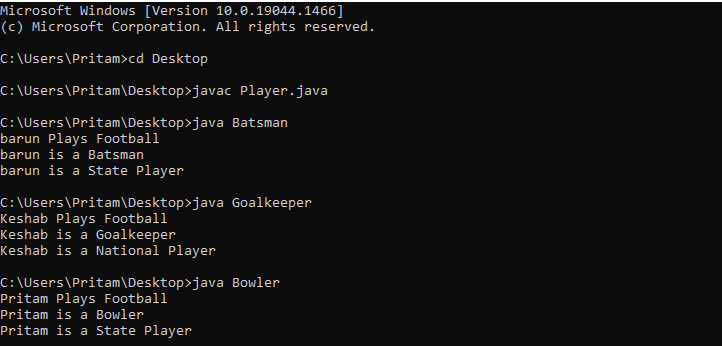
System.***out***.println(sanskar.getName() +" is a Allrounder");

sanskar.team(sanskar.getName());

}

}

**OUTPUT**

****

**Q.N.3)** An organization has different departments and different types of employees. Scientist, Engineers like the software, embedded and hardware engineers, And workers like regular and daily wage workers.  Define their possible properties and methods using appropriate access modifiers. Design possible classes and associate them using inheritance.

* 1. Show how encapsulation concept works for this.
  2. Create objects of each class and access members.
  3. Show how child class can use parent class members.
  4. Write which type(s) of inheritance this solution would take.

Ans.

**public** **class** Employee {

**private** **int** emp\_id, salary;

**private** String name;

**public** **int** getEmp\_id() {

**return** emp\_id;

}

**public** **void** setEmp\_id(**int** emp\_id) {

**this**.emp\_id = emp\_id;

}

**public** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**class** Scientist **extends** Employee{

**void** depart() {

System.***out***.println("The Department is Scientist");

}

**public** **static** **void** main(String[] args) {

Scientist sc=**new** Scientist();

sc.setEmp\_id(101);

sc.setName("Pritam");

sc.setSalary(50000);

System.***out***.println("Name is "+sc.getName()+"\nEmployee\_id is "+sc.getEmp\_id()+"\nand Salary is "+sc.getSalary());

sc.depart();

}

}

**class** Engineers **extends** Employee{

**void** dep() {

System.***out***.println("The Department is Engineer");

}

}

**class** Software **extends** Engineers{

**public** **static** **void** main(String[] args) {

Software so=**new** Software();

so.setEmp\_id(102);

so.setName("keshav");

so.setSalary(40000);

System.***out***.println("Name is "+so.getName()+"\nEmployee\_id is "+so.getEmp\_id()+"\nand Salary is "+so.getSalary());

so.dep();

}

}

**class** Embeded **extends** Engineers{

**public** **static** **void** main(String[] args) {

Embeded em=**new** Embeded();

em.setEmp\_id(103);

em.setName("barun");

em.setSalary(30000);

System.***out***.println("Name is "+em.getName()+"\nEmployee\_id is "+em.getEmp\_id()+"\nand Salary is "+em.getSalary());

em.dep();

}

}

**class** Hardware **extends** Engineers{

**public** **static** **void** main(String[] args) {

Hardware hr=**new** Hardware();

hr.setEmp\_id(104);

hr.setName("barun");

hr.setSalary(40000);

System.***out***.println("Name is "+hr.getName()+"\nEmployee\_id is "+hr.getEmp\_id()+"\nand Salary is "+hr.getSalary());

hr.dep();

}

}

**class** Workers **extends** Employee{

**void** Department() {

System.***out***.println("The department is Workers");

}

}

**class** DailyWage **extends** Workers{

**public** **static** **void** main(String[] args) {

DailyWage dw=**new** DailyWage();

dw.setEmp\_id(105);

dw.setName("Sanskar");

dw.setSalary(20000);

System.***out***.println("Name is "+dw.getName()+"\nEmployee\_id is "+dw.getEmp\_id()+"\nand Salary is "+dw.getSalary());

dw.Department();

}

}

**class** Regular **extends** Workers{

**public** **static** **void** main(String[] args) {

Regular rg=**new** Regular();

rg.setEmp\_id(106);

rg.setName("Manoj");

rg.setSalary(20000);

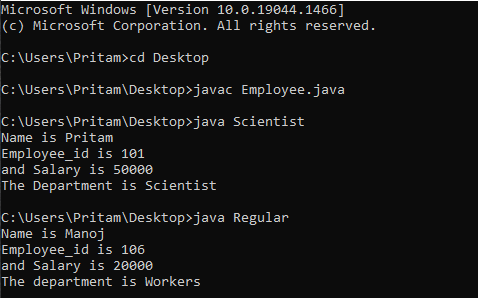
System.***out***.println("Name is "+rg.getName()+"\nEmployee\_id is "+rg.getEmp\_id()+"\nand Salary is "+rg.getSalary());

rg.Department();

}

}

**OUTPUT**

****