1. Write a singletone class.Confirm that singletone class inherited.

Ans: **public** **class** Student {

**static** Student *st*= **new** Student();

**private** Student() {

System.***out***.println();

}

**static** Student getInstance() {

**return** *st*;

}

}

**public** **class** Main {

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

Student st1=**new** Student();

}

}

1. Write a program that describes the hierarchy of an organization.Here we need to write 3 classes Employee,Manager & Labour where Manager & Labour are sub classes of the Employee.Manager has incentive & Labour has over time.Add the functionality to calculate total salary of all the employees.Use polymorphism ie. Method overriding.

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

**int** id,salary;

String name;

Employee(){

name=**null**;

salary=0;

}

Employee(String name,**int** salary){

**this**.name=name;

**this**.salary=salary;

}

**int** getSalary() {

**return** salary;

}

**void** totalsalary(**int** a, **int** b) {

**int** total=a+b;

System.***out***.println("Total salary of all the employees"+total);

}

**void** display() {

**this**.name=name;

**this**.salary=salary;

System.***out***.println("Name of the employee " +name+" Salary is "+salary);

}

**public** **class** Manager **extends** Employee {

Manager(String name,**int** sal) {

**super**(name,sal);

}

**int** getSalary() {

**return**(**super**.getSalary());

}

}

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

**int** c,d;

Employee e=**new** Employee();

Manager m=**new** Manager("Manu",2500);

Labour l=**new** Labour(1000,"Anu");

l.display();

m.display();

c=l.getSalary();

d=m.getSalary();

e.totalsalary(c, d);

}

}

1. Write a program to consider saving & current account in the bank.Saving account holder has ‘Fixed Deposits’ whereas Current account holder has cash credit.Apply polymorphism to find out total cash in the bank

Ans

**public** **class** Bank{

**public** **void** display(**int** a,**int** b) {

**int** total;

total=a+b;

System.***out***.println("TotalCash in bank is "+total);

}

**public** **class** Savings {

**public** **void** display(**int** fd) {

System.***out***.println("Fixed deposit is "+fd);

}

}

**public** **class** CurrentAccount {

**public** **void** display(**int** cashcredit) {

System.***out***.println("Cashcredit "+cashcredit);

}

}

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

**int** fixed, cc ;

CurrentAccount c=**new** CurrentAccount();

Savings s=**new** Savings();

Bank b=**new** Bank();

Scanner sc= **new** Scanner(System.***in***);

/\*

System.out.println("Enter fixed deposit");

fixed=sc.nextInt();

System.out.println("Enter Current cash credit");

cc=sc.nextInt();

\*/

s.display(fixed);

c.display(cc);

b.display(fixed, cc);

}

}

5) Write the classes Line, Rectangle, Cube etc. & make the Shape as their base class. Add an abstract draw()method in the class Shape & draw all shapes

**public** **abstract** **class** Shape {

**abstract** **void** draw();

}

**public** **class** Rectangle **extends** Shape {

**void** draw() {

System.***out***.println("drawing Rectangle");

}

}

**public** **class** Line **extends** Shape{

**void** draw() {

System.***out***.println("drawing Line");

}

}

**public** **class** Circle **extends** Shape {

**void** draw() {

System.***out***.println("drawing Circle");

}

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

Rectangle r=**new** Rectangle();

Line l=**new** Line();

Circle c=**new** Circle();

r.draw();

l.draw();

c.draw();

}

}

6) Write an abstract class ‘Persistence’ along with two sub classes ‘File Persistence’ & ‘DataPersistance’.The base class with have abstract method persist() which will overridden by its sub classes. Write a client who gets the Persistance object at runtime & invokes persist() method on it without knowing whether data is being saved in File or in Database.

Ans

**public** **abstract** **class** Persistance {

**abstract** **void** persist();

}

**public** **class** FilePersistence **extends** Persistance {

**void** persist() {

System.***out***.println("Data Saved in File");

}

}

**public** **class** DataPersistance **extends** Persistance {

**void** persist() {

System.***out***.println("Data saved in Database");

}

}

**public** **class** client **extends** Persistance {

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

Persistance obj=**new** client();

obj.persist();

}

**void** persist() {

System.***out***.println("persist method is invoked by client class");

}

}

1. FOURTH QUESTION

Ans:

**public** **abstract** **class** Base {

Base(){

System.***out***.println("Base Constructor is called");

}

**abstract** **void** fun();

}

**public** **class** Derived **extends** Base {

Derived(){

System.***out***.println("Derived Constructor called");

}

**void** fun() {

System.***out***.println("Derived function is called");

}

}

**public** **class** Main {

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

Derived d=**new** Derived();

}

}

7)SEVENTH QUESTION -DESSERT SHOP APPLICATION

**public** **abstract** **class** DessertItems {

**abstract** **void** getCost();

}

**import** java.util.Scanner;

**public** **class** Icecream **extends** DessertItems {

**public** **void** getCost() {

**double** tax,price=0;

**int** quantity,cost[]= {10,20,30};

tax=1.5;

System.***out***.println("Tax is " +tax);

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter quantity");

quantity=s.nextInt();

price=cost[3]\*quantity\*tax;

System.***out***.println("Cost is"+price);

}

}

**import** java.util.Scanner;

**public** **class** Cookie **extends** DessertItems {

**public** **void** getCost() {

**double** tax,price=0;

tax=1.33;

**int** quantity,cost[]= {10,20,30};

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter quantity");

quantity=s.nextInt();

System.***out***.println("Tax is " +tax);

price=quantity\*tax\*70\*cost;

System.***out***.println("Cost is"+cost);

}

}

**import** java.util.Scanner;

**public** **class** Candy **extends** DessertItems {

**public** **void** getCost() {

**double** tax,price=0;

**int** quantity,cost[]= {10,20,30};

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter quantity");

quantity=s.nextInt();

tax=1.5;

System.***out***.println("Tax is " +tax);

price=quantity\*tax\*60\*cost;

System.***out***.println("Cost is"+cost);

}

}

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** DessertShop {

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

String dessert[]= {"Candy","Cookie","Icecream"};

DessertItems candy=**new** Candy();

DessertItems cookie=**new** Cookie();

DessertItems ice=**new** Icecream();

**int** qty[]= {12,23,45},cost[]= {10,20,30};

**char** ch;

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Choose role");

System.***out***.println("a.Owner\n b. Customer\n");

ch=sc.next().charAt(0);

**if**(ch=='a') {

System.***out***.println("Owner");

System.***out***.println("Dessert to be added");

System.***out***.println("1.Candy, 2.Cookie, 3.Icecream");

System.***out***.println("Choose 1/2/3");

**int** a=sc.nextInt();

**if**(a==1||a==2||a==3) {

System.***out***.println("Enter quantity to be added");

// int total=sc.nextInt();

qty[a-1]=qty[a-1]+sc.nextInt();

System.***out***.println("Dessert added succefully");

}

}

**else** {

System.***out***.println("Customer");

System.***out***.println("Dessert you wanted to add place an order for");

System.***out***.println("1.Candy, 2.Cookie, 3.Icecream");

System.***out***.println("Enter your Item");

**int** a=sc.nextInt();

**switch**(a) {

**case** 1:

System.***out***.println(" Candy ");

System.***out***.println("Quantity : "+qty[a-1]);

System.***out***.println("Price: "+cost[a-1]);

candy.getCost();

System.***out***.println("Your order has been placed");

**break**;

**case** 2:

System.***out***.println(" Cookie ");

System.***out***.println("Quantity : "+qty[a-1]);

System.***out***.println("Price: "+cost[a-1]);

cookie.getCost();

System.***out***.println("Your order has been placed");

**break**;

**case** 3:

System.***out***.println(" Icecream ");

System.***out***.println("Quantity : "+qty[a-1]);

System.***out***.println("Price: "+cost[a-1]);

ice.getCost();

System.***out***.println("Your order has been placed");

**break**;

**default**:

**throw** **new** IllegalArgumentException("Unexpected value: " );

}

}

}

}