Assignment --2

# 1):/\*

# \* Program to calculate area of rectangle using single inheritance in java.

# \*/

package oopsconcept;

class parent

{

int length;

int breadth;

int area;

public void show()

{

length =10;

breadth =5;

area =length\*breadth;

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

}

}

public class assignment41 extends parent {

public static void main(String[] args) {

parent prnt =new parent();

prnt.show();

}

}

2): /\*

\* Program to calculate salary of an employee using single inheritance in java.

\*/

package oopsconcept;

class employee

{

String name;

float basic\_salary;

float increase\_salary;

float bonus;

float total\_salary=0;

public void show()

{

name ="Siddu";

basic\_salary =35000;

System.***out***.println("INCREASING the salary 30 persent of basic salary");

increase\_salary =basic\_salary/30;

bonus =10000;

total\_salary =basic\_salary+increase\_salary+bonus;

System.***out***.println("The total salary of the Employee::"+total\_salary);

}

}

public class assignment42 extends employee {

public static void main(String[] args) {

employee emp =new employee();

emp.show();

}

}

# 3)):/\*

# Program to print library record of student using single inheritance in java.

# \*/

package oopsconcept;

class student

{

String name;

int rollnumber;

public void show()

{

name ="Siddu";

rollnumber=25;

System.***out***.println("Enter the name : " +name);

System.***out***.println("Enter the rollnumber:" +rollnumber);

}

}

public class assignment43 extends student {

String bookname;

String issuedate;

public void display()

{

bookname ="The rules of Life";

issuedate ="03-08-2024";

System.***out***.println("Enter the bookname:" +bookname);

System.***out***.println("enter the issuedate:" +issuedate);

}

public static void main(String[] args) {

assignment43 ass =new assignment43();

ass.show();

ass.display();

}

}

# 4)):/\*

# Single inheritance with methods

# a. Create base class Shape with method draw ()

# b. Create derived class circle that inherits from the shape class.

# c. The Circle class has an additional method draw Circle ().

# d. The program demonstrates calling both methods from the derived class.

# \*/

package oopsconcept;

class shape

{

public void draw()

{

System.***out***.println("DRAWING THE SHAPE");

}

}

class circle extends shape

{

public void drawCircle()

{

System.***out***.println("DRAWING THE CRICLE");

}

}

public class assignment34 {

public static void main(String[] args) {

circle c =new circle();

c.draw();

c.drawCircle();

}

}

# 5)):/\* Single inheritance with Fields

# a. The vehicle class has a field ‘type’ and the car class inherits from it. However,

# car class also has its own field name ‘type’ which hides the field form the base

# class. In this case, when calling displayType () from an instance of Car it accesses the

# field from the base class while displayCarType () accesses the field from the derived class.

# \*

# \*/

package oopsconcept;

class vehicle

{

String type = "Generic Vehicle";

public void displaytype()

{

System.***out***.println("Vehicle Type: " + type);

}

}

class car extends vehicle

{

String type ="BMW";

void displaycarType()

{

System.***out***.println("CAR type:"+type);

}

}

public class assignment35 {

public static void main(String[] args) {

car cr =new car();

cr.displaytype();

cr.displaycarType();

}

}

# 6)):/\*Single Inheritance with Constructor

# \* a. Create a base class Person with parameterized constructor and a method

# \* displayInfo () The derived class student inherits from the Person class and uses the super

# \* keyword to call the constructor of the base class. It also adds its own field rollNumber

# \* and a method displayStudentInfo (). The program demonstrates creating an instance of

# \* student and calling methods from both the base and derived classes.

# \*/

package oopsconcept;

class person

{

String name;

int age;

public person(String name,int age)

{

this.name =name;

this.age =age;

}

public void displayInfo()

{

System.***out***.println("The name of student:"+name);

System.***out***.println("The age of student :"+age);

}

}

class std extends person

{

public std(String name, int age,int rollnumber) {

super(name, age);

this.rollnumber = rollnumber;

}

int rollnumber;

public void displayInfo2()

{

System.***out***.println("Enter the roll number:"+rollnumber);

}

}

public class assignment36 {

public static void main(String[] args) {

std student = new std("Siddu",22,2014);

student.displayInfo();

student.displayInfo2();

}

}

# 7)): /\*

# \* Single Inheritance with Static methods

# a.Create a base class MathUtility with a static method square () that

# calculates the square of an integer. The derived class Calculator inherits from the

# MathUtilitly class adds a static methods cube () that calculates the cube of an integer

# using the square () method from the base class. The program demonstrates calling the static

# methods from the derived class.

# \*/

package oopsconcept;

class MathUtility {

// Static method to calculate the square of an integer

public static int square(int number) {

return number \* number;

}

}

// Derived class

class Calculator extends MathUtility {

// Static method to calculate the cube of an integer using the square method from the base class

public static int cube(int number) {

return number \* *square*(number);

}

}

public class assignment37 {

public static void main(String[] args) {

int number =4;

int squareResult = Calculator.*square*(number);

int cubeResult = Calculator.*cube*(number);

System.***out***.println("Square of " + number + " is: " + squareResult);

System.***out***.println("Cube of " + number + " is: " + cubeResult);

}

}

# 8)):/\*

# \*Create three classes: Animal, Dog, and Labrador. The Animal class is the base class

# \*that has a method eat(). The Dog class is the child class that inherits from Animal and

# \*adds a new method bark(). The Labrador class is the grandchild class that inherits from

# \*Dog and adds a new method displayColor().

# \*When we create an instance of the Labrador class

# \*and call its methods, it can access all the methods from its base class and its parent

# 

# \*/

package oopsconcept;

class Animal

{

public void eat()

{

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

}

}

class Dog extends Animal

{

public void bark()

{

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

}

}

class Labrador extends Dog

{

public void displayColor()

{

System.***out***.println("BLACK COLOR");

}

}

public class assignment38 {

public static void main(String[] args) {

Labrador Lbr =new Labrador();

Lbr.eat();

Lbr.bark();

Lbr.displayColor();

}

}

# 9)): Create three classes: Vehicle, Car, and SportsCar. The Vehicle class is the base class that has a method start(). The Car class is the child class that inherits from Vehicle and adds a new method drive(). The SportsCar class is the grandchild class that inherits from Car and adds a new method accelerate(). When we create an instance of the SportsCar class and call its methods, it can access methods from all three levels of the inheritance hierarchy.

package oopsconcept;

class Vehicle

{

public void Start()

{

System.***out***.println("Vehicle is Starts");

}

}

class Car extends Vehicle

{

public void drive()

{

System.***out***.println("Car is Starting");

}

}

class SportsCar extends Car

{

public void accelerate()

{

System.***out***.println("SportsCar is accelerate");

}

}

public class assignment310 {

public static void main(String[] args) {

SportsCar SC =new SportsCar();

SC.Start();

SC.drive();

SC.accelerate();

}

}

# 10)): Create three classes: Shape, Circle, and ColoredCircle. The Shape class is the base class that has a method draw(). The Circle class is the child class that inherits from Shape and adds a new method drawCircle(). The ColoredCircle class is the grandchild class that inherits from Circle and adds a new method setColor(String color). When we create an instance of the ColoredCircle class and call its methods, it can access methods from all three levels of the inheritance hierarchy.

package oopsconcept;

class shape

{

public void draw()

{

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

}

}

class Cricle extends shape

{

public void drawCricle()

{

System.***out***.println("DRAWING THE CRICLE");

}

}

class coloredCricle extends Cricle

{

String color;

public void setColor(String color)

{

this.color =color;

System.***out***.println("Coloring the circle with " + color);

}

}

public class assignment311 {

public static void main(String[] args) {

coloredCricle col =new coloredCricle();

col.draw();

col.drawCricle();

col.setColor("red");

}

}

# 11)): /\* Create three classes: Person, Employee, and Manager.

# \* The Person class is the base class that has a constructor to set the person's

# \* name and a method displayInfo() to display the name. The Employee class is the child class that inherits

# \* from Person and adds a salary field with a constructor to set both the name and salary, and a method displaySalary() to display the salary.

# \* The Manager class is the grandchild class that inherits from Employee and adds a department field with a constructor to set all three attributes,

# \* and a method displayDepartment() to display the department. When we create an instance of the Manager class and call its methods, it can access methods

# \* and fields from all three levels of the inheritance hierarchy.

# \*/

package oopsconcept;

class person

{

String name;

int age;

String date\_of\_birth;

public void displayInfo(String name,int age,String date\_of\_birth)

{

this.name =name;

this.age =age;

this.date\_of\_birth =date\_of\_birth;

}

public void displayInfo() {

System.***out***.println("Name: " + name);

System.***out***.println("Age: " + age);

System.***out***.println("Date of Birth: " + date\_of\_birth);

}

}

class Employee extends person

{

int emp\_id;

int salary;

public void displaySalary(int emp\_id,int salary)

{

this.emp\_id =emp\_id;

this.salary =salary;

}

public void displaySalary() {

System.***out***.println("Employee ID: " + emp\_id);

System.***out***.println("Salary: " + salary);

}

}

class manager extends Employee

{

public String Technical\_skill="java";

String Techincal\_skill;

public void displayDepartment(String Techincal\_skill)

{

this.Techincal\_skill =Techincal\_skill;

}

public void displayDepartment() {

System.***out***.println("Technical Skill: " + Technical\_skill);

}

}

public class assignment312 extends manager {

public static void main(String[] args) {

manager mng =new manager();

mng.displaySalary(123,20000);

mng.displayInfo("SIDDU",21,"25-07-2002");

mng.displayDepartment();

mng.displayInfo();

mng.displaySalary();

mng.displayDepartment();

}

}