|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  **Topic: Inheritance** | **Subject:**  **Object Oriented Programming**  **submitted by:**  **Daoud hussain**  (Sp21-bcs-102)  **submitted to:**  **Mam Saneeha AAmir**  **date of submission:**  **April 10 , 2022** |

1. Computer Class:

public class Computer{

private int wordSize, memorySize;

private int storageSize, speed;

public Computer(){

//Default Constructor

}

//Full-Argument Constructor

public Computer(int ws, int ms, int ss, int s){

if(ws!=0 && ms!=0 && ss!=0 && s!=0 ){

wordSize = ws;

memorySize = ms;

storageSize = ss;

speed = s;

}

}

//Method to display Parent class data

public void display(){

System.out.println("Word Size: " + wordSize + "wpm");

System.out.println("Memory: " + memorySize + "GB");

System.out.println("Storage: " + storageSize + "GB");

System.out.println("Speed: " + speed + "mbps");

}

}

1. Laptop Class:

public class Laptop extends Computer{

private int length, width;

private int height, weight;

public Laptop(){

//Default Constructor

}

//Full-Argument Constructor

public Laptop(int ws, int ms, int ss, int s,int l, int w, int h, int we){

super(ws, ms, ss, s);

if(l!=0 && w!=0 && h!=0 && we!=0){

length = l;

width = w;

height = h;

weight = we;

}

}

//Method to display Child and Parent class data

public void display(){

//Overriding base class method

super.display();

System.out.println("Length: " + length + "m");

System.out.println("Width: " + width + "m");

System.out.println("Height: " + height + "m");

System.out.println("Weight: " + weight + "kg");

}

}

1. Laptop Runner:

public class Runner{

public static void main(String[] args) {

//Making child object

Laptop myLaptop = new Laptop(12, 8, 512, 20, 10, 10, 8, 5);

myLaptop.display();

}

}

-------------------------------

1. Commision-Employee Class:

public class CommisionEmployee{

String firstName;

String lastName;

String ssl;

double grossSales;

double commisionRate;

public CommisionEmployee(){

//Default Argument Constructor

}

//Full-Argument Constructor

public CommisionEmployee(String fn, String ln, String sslArg, double gs, double cr){

firstName = fn;

lastName = ln;

ssl = sslArg;

grossSales = gs;

commisionRate = cr;

}

//Method to display Parent class data

public void display(){

System.out.println("First Name: " + firstName);

System.out.println("Last Name: " + lastName);

System.out.println("SSL: " + ssl);

System.out.println("Gross Sales: " + grossSales);

System.out.println("Commision Rate: " + commisionRate);

}

//Method to calculate Parent class earning

public double calculateEarning(){

double earning = grossSales\*commisionRate;

return earning;

}

}

1. Base-Plus Commision Employee Class:

public class BasePlusCommisionEmployee extends CommisionEmployee{

double salary;

//Child Class Object

public BasePlusCommisionEmployee(String fn, String ln, String ssl, double gs, double cr ,double sal){

super(fn,ln,ssl,gs,cr);

salary = sal;

}

//Reseting Value of Salary in child class

void setSalary(double sal){

if(sal != 0){

salary = sal;

}

}

//Geting Value of Salary from child class

double getSalary(){

return salary;

}

//Method to display Child class data

public void display(){

super.display();

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

System.out.println("Earning: " + calculateEarning());

}

//Method to calculate Child class earning that is parent earning + child salary

public double calculateEarning(){

return super.calculateEarning()+salary;

}

}

1. Base-Plus Commision Employee Runner:

public class Runner{

public static void main(String[] args) {

BasePlusCommisionEmployee bpce1 = new BasePlusCommisionEmployee("Daoud","Hussain","SP21",20.5,3.4,20000);

bpce1.setSalary(25000);

bpce1.display();

bpce1.calculateEarning();

}

}

-------------------------------

1. Publication Class:

public class Publication{

private int price;

private String title;

//Setter Methods

public void setPrice(int p){

if(p != 0){

price = p;

}

}

public void setTitle(String t){

if(t != ""){

title = t;

}

}

//Getter Methods

public int getPrice(){

return price;

}

public String getTitle(){

return title;

}

public void display(){

System.out.println("Price: " + price);

System.out.println("Title: " + title);

}

}

1. Book Class:

public class Book extends Publication{

private int pageCount;

//Setter Method

public void setPageCount(int pc){

if(pc != 0){

pageCount = pc;

}

}

//Getter Method

public int getPageCount(){

return pageCount;

}

public void display(){

System.out.println("Price: " + getPrice());

System.out.println("Title: " + getTitle());

System.out.println("Page Counts: " + pageCount);

}

}

1. Tape Class:

public class Tape extends Publication{

private int playingTimeInMinutes;

//Setter Method

public void setPlayingTimeInMinutes(int ptim){

if(ptim != 0){

playingTimeInMinutes = ptim;

}

}

//Getter Method

public int getPlayingTimeInMinutes(){

return playingTimeInMinutes;

}

public void display(){

System.out.println("Price: " + getPrice());

System.out.println("Title: " + getTitle());

System.out.println("Playing Time: " + playingTimeInMinutes);

}

}

1. Tape and Book Runner:

public class Runner{

public static void main(String[] args) {

//Creating object for Book cLass

Book myBook = new Book();

myBook.setPageCount(30);

myBook.setPrice(250);

myBook.setTitle("Introduction to Java");

myBook.display();

System.out.println();

//Creating object for Tape cLass

Tape myTape = new Tape();

myTape.setPlayingTimeInMinutes(60);

myTape.setPrice(150);

myTape.setTitle("Java For Beginners");

myTape.display();

}

}

-------------------------------

1. Simple Class:

public class Simple{

int num1;

int num2;

public int add(){

return num1+num2;

}

public int sub(){

return num1-num2;

}

public int mul(){

return num1\*num2;

}

public int div(){

return num1/num2;

}

}

1. Verified-Simple Class:

public class VerifiedSimple extends Simple{

public int add(){

if(num1 > 0 && num2 >0){

return num1+num2;

}

else{

System.out.println("Error! Addition function will be evaluated iff both values will be greater than 0");

return -1;

}

}

public int sub(){

if(num1 > 0 && num2 >0){

return num1-num2;

}

else{

System.out.println("Error! Subtract function will be evaluated iff both values will be greater than 0");

return -1;

}

}

public int mul(){

if(num1 > 0 && num2 >0){

return num1+num2;

}

else{

System.out.println("Error! Multiply function will be evaluated iff both values will be greater than 0");

return -1;

}

}

public int div(){

if(num1 > 0 && num2 >0){

return num1/num2;

}

else{

System.out.println("Error! Division function will be evaluated iff both values will be greater than 0");

return -1;

}

}

}

4. Verified-Simple Runner

public class Runner{

public static void main(String[] args) {

VerifiedSimple vs = new VerifiedSimple();

vs.add();

vs.sub();

vs.mul();

vs.div();

}

}

-------------------------------

5. Person Class:

public class Person{

int age;

String name;

public Person(){

//Default Constructor

}

//Full-Argument Constructor

public Person(int a, String n){

if(a > 0 && n!=""){

name = n;

age = a;

}

}

public int getAge(){

return age;

}

public String getName(){

return name;

}

}

5. Vehicle Class

public class Vehicle{

String manufacturerName;

int numberOfCylinders;

Person owner;

public Vehicle(){

Person p1 = new Person();

}

public Vehicle( String mn, int nc, Person p1){

if(mn != "" && nc !=0){

manufacturerName = mn;

numberOfCylinders = nc;

owner = p1;

}

}

}

5. Truck class

public class Truck extends Vehicle{

double capacityInTons;

int capacityInPounds;

Person per1 = new Person(20, "Daoud");

public Truck(String mn, int nc, Person per1, int cp, double ct){

super(mn, nc, per1);

if(cp != 0 && ct != 0){

capacityInPounds = cp;

capacityInTons = ct;

}

}

public void display(){

System.out.println("Owner Name: " + per1.getName());

System.out.println("Owner Age: " + per1.getAge());

System.out.println("Manufacture's Name: " + manufacturerName);

System.out.println("Number of Cylinders: " + numberOfCylinders);

System.out.println("Load Capacity in Tons: " + capacityInTons);

System.out.println("Load Capacity in Pounds: " + capacityInPounds);

}

}

5. Truck and Vehicle Runner

public class Runner{

public static void main(String[] args) {

Person p1 = new Person();

Truck myTruck = new Truck("Honda", 25, p1, 4000, 3578.9);

myTruck.display();

}

}

-------------------------------

6. Person Class

public class Person2{

private String name, email;

private String address, phoneNumber;

//Setters

public void setName(String n){

if(n!=""){

name = n;

}

}

public void setAddress(String n){

if(n!=""){

address = n;

}

}

public void setEmail(String n){

if(n != "" && n.contains("@")){

email = n;

}

}

public void setPhoneNumber(String n){

if(n!="" && n.length() == 11){

phoneNumber = n;

}

}

public Person2(String a, String n, String p, String e){

if(a != "" && n != "" && p!="" && e != ""){

email = e;

address = a;

phoneNumber = p;

name = n;

}

}

//Getters

public String getName(){

return name;

}

public String getAddress(){

return address;

}

public String getPhoneNumber(){

return phoneNumber;

}

public String getEmail(){

return email;

}

}

6. Employee Class

public class Employee extends Person2{

private double salary;

private int office;

private Date hireDate;

Date hd = new Date();

public Employee(String a, String n, String p, String e,double s, int o, Date hd){

super(a, n, p, e);

salary = s;

office = o;

hireDate = hd;

}

public Date getHireDate()

{

return hireDate;

}

public double getSalary()

{

return salary;

}

public int getOffice()

{

return office;

}

}

6. Date Class

public class Date{

private int day;

private int month;

private int year;

Date(){

//Default Argument Constructor

}

Date(int h, int s, int c){

day = h;

month = s;

year = c;

}

public int getDay(){

return day;

}

public int getMonth(){

return month;

}

public int getYear(){

return year;

}

}

6. Faculty Class

public class Faculty extends Employee{

private int officeHours;

private int ranks;

Date hiredDate = new Date(12,04,2022);

public Faculty(String a, String n, String p, String e,double s, int o, Date hd, int oh, int r){

super(a, n, p, e, s, o, hd);

officeHours = oh;

ranks = r;

}

public void display(){

System.out.println("Person Name: " + getName());

System.out.println("Person Address: " + getAddress());

System.out.println("Person Phone-No: " + getPhoneNumber());

System.out.println("Person Email: " + getEmail());

System.out.println("Employee Office: " + getOffice());

System.out.println("Employee Salary: " + getSalary());

System.out.println("Employee Date: " + hiredDate.getDay()+ "-" + hiredDate.getMonth()+ "-" + hiredDate.getYear());

System.out.println("Faculty Office Hours: " + officeHours);

System.out.println("Ranks: " + ranks);

}

}

6. Staff Class

public class Staff extends Employee{

private String title;

Date hiredDate = new Date(31,3,2022);

public Staff(String a, String n, String p, String e,double s, int o, Date hd, String t){

super(a, n, p, e, s, o, hd);

title = t;

}

public void display(){

System.out.println("Person Name: " + getName());

System.out.println("Person Address: " + getAddress());

System.out.println("Person Phone-No: " + getPhoneNumber());

System.out.println("Person Email: " + getEmail());

System.out.println("Employee Office: " + getOffice());

System.out.println("Employee Salary: " + getSalary());

System.out.println("Employee Date: " + hiredDate.getDay()+ "-" + hiredDate.getMonth()+ "-" + hiredDate.getYear());

System.out.println("Staff Title: " + title);

}

}

6. Runner Staff and Faculty Classes

public class Runner{

public static void main(String[] args) {

Date todayDate = new Date();

Faculty myFaculty = new Faculty("Street-5 Hostel City Islamabad", "Daoud", "03483016704", "daoudhussain302@gmail.com",25000.5, 1, todayDate, 9,5);

System.out.println("FACULTY DATA");

System.out.println("---------------------------------------------------");

myFaculty.display();

System.out.println();

System.out.println("STAFF DATA");

System.out.println("---------------------------------------------------");

Staff myStaff = new Staff("Park-Road Tarlai Kalan Islamabad", "Rizwan", "03442234426", "rizwan@comsats.edu.pk",175000.5, 1, todayDate,"Teacher");

myStaff.display();

System.out.println("---------------------------------------------------");

}

}

-------------------------------