|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  OBJECT ORIENTED PROGRAMMING  *Lab Task 08*  *Abstraction and polymorphism* | **submitted by:**  **Shahzaneer Ahmed**  **registration number:**  **sp21-bcs-087**  **submitted to:**  **mA’M sANEEHA aMIR**  **date of submission:**  **APRIL 23, 2022** |

Question 1

# Source Code

## Convert

public abstract class Convert{

    private double value1;

    private double value2;

    public Convert() {

    }

    public Convert(double value1) {

*this*.value1 = value1;

    }

    public double getValue1() {

        return *this*.value1;

    }

    public void setValue1(double value1) {

*this*.value1 = value1;

    }

    public double getValue2() {

        return *this*.value2;

    }

    public void setValue2(double value2) {

*this*.value2 = value2;

    }

    @Override

    public String toString() {

        return "{" +

            " value1='" + getValue1() + "'" +

            ", value2='" + getValue2() + "'" +

            "}";

    }

    public abstract double compute();

}

## FahrenheitToCelcius

public class FahrenheitToCelsius extends Convert {

    public FahrenheitToCelsius(double value1) {

*this*.setValue1(value1);

    }

    @Override

    public double compute() {

        double convertedValue = (((*super*.getValue1() - 32) \* 5) / 9);

        System.out.println(getValue1());

*this*.setValue2(convertedValue);

        return *this*.getValue2();

    }

    @Override

    public String toString() {

        return "The value is : " + *super*.getValue1() +

                "The converted Value is " + *super*.getValue2();

    }

}

## FeetsToMeters

public class FeetsToMeters extends Convert {

    public FeetsToMeters(double value1) {

*this*.setValue1(value1);

    }

    @Override

    public double compute() {

        double convertedValue = *super*.getValue1() \* 0.3048;

*this*.setValue2(convertedValue);

        return *this*.getValue2();

    }

    @Override

    public String toString() {

        return "The value is : " + *super*.getValue1() +

                "The converted Value is " + *super*.getValue2();

    }

}

## LitersToGallons

public class LitersTOGallons extends Convert {

    public LitersTOGallons(double value1) {

*this*.setValue1(value1);

    }

    @Override

    public double compute() {

        double convertedValue = *this*.getValue1() \* 0.264172;

*this*.setValue2(convertedValue);

        return *this*.getValue2();

    }

    @Override

    public String toString() {

        return "The value is : " + *super*.getValue1() +

                "The converted Value is " + *super*.getValue2();

    }

}

## Runner

public class Runner {

    public static void main(String[] args) {

        Convert conversions[] = new Convert[3];

        conversions[0] = new FahrenheitToCelsius(23);

        conversions[1] = new FeetsToMeters(12);

        conversions[2] = new LitersTOGallons(10);

        for (int i = 0; i < conversions.length; i++) {

            conversions[i].compute();

            System.out.println(conversions[i].toString());

        }

    }

}

*// Runtime polymorphism --> only when early binding and late binding both are present!*

Question 2

# Source Code

## Employee

public abstract class Employee{

    private String firstName;

    private String lastName;

    private long SSN;

*// Constructors*

    public Employee() {

    }

    public Employee(String firstName, String lastName, long SSN) {

*this*.firstName = firstName;

*this*.lastName = lastName;

*this*.SSN = SSN;

    }

    public abstract double earnings();

    public void display() {

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

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

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

    }

    @Override

    public String toString() {

        return String.format("First name : %s \nLast Name : %s \nSSN : %d", firstName, lastName, SSN);

    }

}

## CommissionEmployee

public class CommisionEmployee extends Employee {

    private int sales;

    private int commissionRate;

    public CommisionEmployee() {

    }

    public CommisionEmployee(String firstName, String lastName, long SSN, int sales, int commissionRate) {

*super*(firstName, lastName, SSN);

*this*.commissionRate = commissionRate;

*this*.sales = sales;

    }

    @Override

    public double earnings() {

        return sales \* commissionRate;

    }

    @Override

    public String toString() {

        return *super*.toString() + String.format("\n Sales : %d\nCommission Rate : %d", sales,commissionRate);

    }

}

## HourlyEmployee

public class HourlyEmployee extends Employee {

    private int hours;

    private double wagePerHour;

    public HourlyEmployee() {

    }

    public HourlyEmployee(String firstName, String lastName, long SSN, int hours, double wagePerHour) {

*super*(firstName, lastName, SSN);

*this*.hours = hours;

*this*.wagePerHour = wagePerHour;

    }

    @Override

    public double earnings() {

        return hours \* wagePerHour;

    }

    @Override

    public String toString() {

        return *super*.toString() + String.format("\nHours : %d\nWage per Hour : %f", hours,wagePerHour);

    }

}

## WeeklyEmployee

public class WeeklyEmployee extends Employee {

    private int weeklySalary;

    public WeeklyEmployee() {

    }

    public WeeklyEmployee(String firstName, String lastName, long SSN, int weeklySalary) {

*super*(firstName, lastName, SSN);

*this*.weeklySalary = weeklySalary;

    }

    @Override

    public double earnings() {

        return weeklySalary;

    }

    @Override

    public String toString() {

        return *super*.toString() + String.format("\nWeekly Salary : %d ", weeklySalary);

    }

}

## BasePlusComissionEmployee

public class BasePlusCommisionEmployee extends CommisionEmployee {

*// basePlusCommissionEmployee extends CommisionEmployee , Employee is not allowed in java*

*// you can only extend one class.*

*// extending two classes means you are doing multiple inheritance which is not allowed in java*

*// however you can implement multiple interfaces.*

    private double basicSalary;

    public BasePlusCommisionEmployee() {

    }

    public BasePlusCommisionEmployee(String firstName, String lastName, long SSN , int sales, int commissionRate, double basicSalary) {

*super*(firstName, lastName, SSN, sales, commissionRate);

*this*.basicSalary = basicSalary;

    }

    @Override

    public double earnings() {

        return *super*.earnings() + basicSalary;

    }

    public void setBaseSalary(double basicSalary) {

*this*.basicSalary = basicSalary;

    }

    public double getBaseSalary() {

        return *this*.basicSalary;

    }

    @Override

    public String toString() {

        return *super*.toString() + String.format("\nBasic Salary : %f", basicSalary);

    }

}

## Runner

public class Runner {

    public static void main(String[] args) {

*// ! This is polymorphism --> as we have used the generic code to call all other instances*

        Employee employees[] = new Employee[4];

*//\* Yeh hai jo hum normally krte hain!*

*// HourlyEmployee e1 = new HourlyEmployee(2, 500);*

*// WeeklyEmployee e2 = new WeeklyEmployee(7000);*

*// CommisionEmployee e3 = new CommisionEmployee(7, 1700);*

*// BasePlusCommisionEmployee e4 = new BasePlusCommisionEmployee(5000);*

*//?  objects ko employee class ki array main rkna hai*

*//? yaad rahay k Abstract class ka object nhi bnta lekin yeh aik type hai*

        employees[0] = new WeeklyEmployee("Maleeka", "Naqvi", 2345, 7000);

        employees[1] = new HourlyEmployee("Sameem", "Amjad", 25689, 18, 567);

        employees[2] = new CommisionEmployee("Laiba", "Imran", 4532, 45, 1900);

        employees[3] = new BasePlusCommisionEmployee("Raheem", "Arif", 5609, 12, 1300, 7000);

*//\* display ki jageh toString ko override karen.*

        for (int i = 0; i < employees.length; i++) {

*// earnings pehle resolve hua hai parent main (early binding and then yeh resolve hua hai child main late binding aur in dono*

*//ki wajeh se yeh polymorphic code hai toString bhi directly inherit tha Object class se )*

            System.out.println(employees[i].earnings());

            System.out.println(employees[i].toString());

*// DownCasting*

*// ager hum change krna chahain BaseCommissionEmployee ki salary tou hamain isko downcast kr k krna hoga q k setSalary method*

*//Employee class (parent) k pas nhi hai.*

*// Downcasting ki wajeh se original definition nhi alter krni pari hamain.*

            if (employees[i] instanceof BasePlusCommisionEmployee) {

                BasePlusCommisionEmployee emp = (BasePlusCommisionEmployee) employees[i];

                emp.setBaseSalary(1.10 \* emp.getBaseSalary());

                System.out.println("new base salary with 10%% increase is "+ emp.getBaseSalary());

                employees[i] = emp;

            }

        }

    }

}

Question 3

# Source Code

## Action

import java.util.Scanner;

public class Action extends Movie {

    private double lateFee;

    public double getLateFee() {

        return *this*.lateFee;

    }

    public void setLateFee(double lateFee) {

*this*.lateFee = lateFee;

    }

    public Action(int numberID, String title, int numberOfDaysRented) {

*super*(numberID, title, numberOfDaysRented);

*this*.lateFee = 3.0;

    }

    @Override

    public double calcLateFee() {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter the Number of late days : ");

        int days = input.nextInt();

        return (*this*.lateFee \* days);

    }

    @Override

    public String toString() {

        return *super*.toString() +

            " lateFee='" + getLateFee() + "'" +

            "}";

    }

}

## Comedy

import java.util.Scanner;

public class Comedy extends Movie {

    private double lateFee;

    public double getLateFee() {

        return *this*.lateFee;

    }

    public void setLateFee(double lateFee) {

*this*.lateFee = lateFee;

    }

    public Comedy(int numberID, String title, int numberOfDaysRented) {

*super*(numberID, title, numberOfDaysRented);

*this*.lateFee = 2.5;

    }

    @Override

    public double calcLateFee() {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter the Number of late days : ");

        int days = input.nextInt();

        return (*this*.lateFee \* days);

    }

    @Override

    public String toString() {

        return *super*.toString() +

            " lateFee='" + getLateFee() + "'" +

            "}";

    }

}

## Drama

import java.util.Scanner;

public class Drama extends Movie {

    private double lateFee;

    public double getLateFee() {

        return *this*.lateFee;

    }

    public void setLateFee(double lateFee) {

*this*.lateFee = lateFee;

    }

    public Drama(int numberID, String title, int numberOfDaysRented) {

*super*(numberID, title, numberOfDaysRented);

*this*.lateFee = 2.0;

    }

    @Override

    public double calcLateFee() {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter the Number of late days : ");

        int days = input.nextInt();

        return (*this*.lateFee \* days);

    }

    @Override

    public String toString() {

        return *super*.toString() +

            " lateFee='" + getLateFee() + "'" +

            "}";

    }

}

## Movie

public abstract class Movie {

    private int numberID;

    private String title;

    private int numberOfDaysRented;

    public Movie() {

    }

    public Movie(int numberID, String title, int numberOfDaysRented) {

*this*.numberID = numberID;

*this*.title = title;

*this*.numberOfDaysRented = numberOfDaysRented;

    }

    public int getNumberID() {

        return *this*.numberID;

    }

    public void setNumberID(int numberID) {

*this*.numberID = numberID;

    }

    public String getTitle() {

        return *this*.title;

    }

    public void setTitle(String title) {

*this*.title = title;

    }

    public int getNumberOfDaysRented() {

        return *this*.numberOfDaysRented;

    }

    public void setNumberOfDaysRented(int numberOfDaysRented) {

*this*.numberOfDaysRented = numberOfDaysRented;

    }

    public boolean equals(Movie that) {

*// It checks the content of two objects only*

        if (*this*.numberID == that.numberID && *this*.title == that.title

                && *this*.numberOfDaysRented == that.numberOfDaysRented) {

            return true;

        }

        return false;

    }

*//! Definition from the extension*

*// if (o == this)*

*//     return true;*

*// if (!(o instanceof Movie)) {*

*//     return false;*

*// }*

*// Movie movie = (Movie) o;*

*// return numberID == movie.numberID && Objects.equals(title, movie.title) && numberOfDaysRented == movie.numberOfDaysRented;*

    @Override

    public String toString() {

        return "{" +

                " numberID='" + getNumberID() + "'" +

                ", title='" + getTitle() + "'" +

                ", numberOfDaysRented='" + getNumberOfDaysRented() + "'" +

                "}";

    }

    public abstract double calcLateFee();

}

## Runner

public class Runner {

    public static void main(String[] args) {

        Movie movies[] = new Movie[3];

        movies[0] = new Action(23, "survival of the fittest", 22);

        movies[1] = new Drama(21, "Ehd e wafa", 100);

        movies[2] = new Comedy(1, "Pakistani democrary", 500);

        for (int i = 0; i < movies.length; i++) {

            System.out.println("Before Downcasting");

            System.out.println(movies[i].toString());

            System.out.println(movies[i].calcLateFee());

*// todo: Reset numberID for action:Movie*

            if (movies[i] instanceof Action) {

                Action newM1 = (Action) movies[i];

                newM1.setNumberID(2);

                movies[i] = newM1;

            }

            System.out.println("After DownCasting ");

            System.out.println(movies[i].toString());

            System.out.println(movies[i].calcLateFee());

        }

*//*

*// System.out.println(movies[0].equals(movies[2]));*

*// System.out.println(movies[0].equals(movies[2]));*

        if (movies[0].getClass() == movies[2].getClass()) {

            System.out.println("Types are same");

*// yeh isliay nhi chala q k polymorphic call hamesha return late binding se hoti hai aur late binding*

*// k level per dono objects different classes k hain!*

        }

    }

}

Question 4

# Source Code

## Ounce

public class Ounce {

    private double weight;

    private float weightInOunce;

*// Whenever we use final with any field it becomes required in java*

*// Dart main hamaray pas required keyword hai*

    public Ounce() {

    }

    public Ounce(float weightInOunce) {

        if(weightInOunce>=0) *this*.weightInOunce = weightInOunce;

    }

    public Ounce(double  weight) {

        if(weight>=0) *this*.weight = weight \* 35.274;

    }

    public double getWeight() {

        return *this*.weight;

    }

    public void setWeight(float weight) {

*this*.weight = weight;

    }

    public float getWeightInOunce() {

        return *this*.weightInOunce;

    }

    public void setWeightInOunce(float weightInOunce) {

*this*.weightInOunce = weightInOunce;

    }

*// generated by java Entension*

*// @Override*

*// public boolean equals(Object o) {*

*//     if (o == this)*

*//         return true;*

*//     if (!(o instanceof Ounce)) {*

*//         return false;*

*//     }*

*//     Ounce ounce = (Ounce) o;*

*//     return weight == ounce.weight && weightInOunce == ounce.weightInOunce;*

*// }*

*// @Override*

*// public int hashCode() {*

*//     return Objects.hash(weight, weightInOunce);*

*// }*

*// @Override*

*// public String toString() {*

*//     return "{" +*

*//             " weight='" + getWeight() + "'" +*

*//             ", weightInOunce='" + getWeightInOunce() + "'" +*

*//             "}";}*

}

## Package

public abstract class Package {

    private String senderName;

    private String senderAddress;

    private String recieverName;

    private String recieverAddress;

    private Ounce weight;

    private double costPerOunce;

    public Package() {

    }

    public Package(String senderName, String senderAddress, String recieverName, String recieverAddress, Ounce weight, double costPerOunce) {

*this*.senderName = senderName;

*this*.senderAddress = senderAddress;

*this*.recieverName = recieverName;

*this*.recieverAddress = recieverAddress;

*this*.weight = weight;

*this*.costPerOunce = costPerOunce;

    }

    public String getSenderName() {

        return *this*.senderName;

    }

    public void setSenderName(String senderName) {

*this*.senderName = senderName;

    }

    public String getSenderAddress() {

        return *this*.senderAddress;

    }

    public void setSenderAddress(String senderAddress) {

*this*.senderAddress = senderAddress;

    }

    public String getRecieverName() {

        return *this*.recieverName;

    }

    public void setRecieverName(String recieverName) {

*this*.recieverName = recieverName;

    }

    public String getRecieverAddress() {

        return *this*.recieverAddress;

    }

    public void setRecieverAddress(String recieverAddress) {

*this*.recieverAddress = recieverAddress;

    }

    public Ounce getWeight() {

        return *this*.weight;

    }

    public void setWeight(Ounce weight) {

*this*.weight = weight;

    }

    public double getCostPerOunce() {

        return *this*.costPerOunce;

    }

    public void setCostPerOunce(double costPerOunce) {

*this*.costPerOunce = costPerOunce;

    }

*// forcibly imposing this method to the childs of this class*

    public abstract double calculateCost();

    public double calculateGeneralCost() {

        return (weight.getWeight() \* costPerOunce);

    }

    @Override

    public String toString() {

        return "{" +

            " senderName='" + getSenderName() + "'" +

            ", senderAddress='" + getSenderAddress() + "'" +

            ", recieverName='" + getRecieverName() + "'" +

            ", recieverAddress='" + getRecieverAddress() + "'" +

            ", weight='" + getWeight().getWeight() + "'" +

            ", costPerOunce='" + getCostPerOunce() + "'" +

            "}";

    }

}

## TwoDaysPackage

public class TwoDayPackage extends Package {

    private double flatFee;

    public TwoDayPackage(String senderName, String senderAddress, String recieverName, String recieverAddress,

            Ounce weight, double costPerOunce, double flatFee) {

*super*(senderName, senderAddress, recieverName, recieverAddress, weight, costPerOunce);

*this*.flatFee = flatFee;

    }

    public double getFlatFee() {

        return *this*.flatFee;

    }

    public void setFlatFee(double flatFee) {

*this*.flatFee = flatFee;

    }

    @Override

    public double calculateCost() {

        return *super*.calculateGeneralCost() + flatFee;

    }

    @Override

    public String toString() {

        return *super*.toString() +

            " flatFee='" + getFlatFee() + "'" +

            "}";

    }

}

## OverNightPackage

public class OverNightPackage extends Package {

    private double additionalFee;

    public OverNightPackage(String senderName, String senderAddress, String recieverName, String recieverAddress,

            Ounce weight, double costPerOunce, double additionalFee) {

*super*(senderName, senderAddress, recieverName, recieverAddress, weight, costPerOunce);

*this*.additionalFee = additionalFee;

    }

    public double getAdditionalFee() {

        return *this*.additionalFee;

    }

    public void setAdditionalFee(double additionalFee) {

*this*.additionalFee = additionalFee;

    }

    @Override

    public double calculateCost() {

        return *super*.calculateGeneralCost() + additionalFee;

    }

    @Override

    public String toString() {

        return *super*.toString()+

            " additionalFee='" + getAdditionalFee() + "'" +

            "}";

    }

}

## Runner

public class Runner {

    public static void main(String[] args) {

        Package pkg[] = new Package[2];

*// father class ki array;*

        Ounce o1 = new Ounce(2.5);

        Ounce o2 = new Ounce(5.0);

        pkg[0] = new OverNightPackage("Shahzaneer Ahmed", "Pakistan", "Hafsa", "Canada", o1, 500, 1000);

        pkg[1] = new TwoDayPackage("Shahzaneer Ahmed", "New Zealand", "Saad Hamid", "Singapore", o2, 700, 2000);

        for (int i = 0; i < pkg.length; i++) {

            System.out.println(pkg[i].toString());

            System.out.println(pkg[i].calculateCost());

        }

    }

}

Question 5

# Source Code

## Person

public abstract class Person{

    private String name;

    public Person() {

    }

    public Person(String name){

*this*.name = name;

    }

    public String getName() {

        return *this*.name;

    }

    public void setName(String name) {

*this*.name = name;

    }

    public abstract boolean isOutstanding();

    @Override

    public String toString() {

        return "{" +

            " name='" + getName() + "'" +

            "}";

    }

}

## Professor

public class Professor extends Person {

    private int numberOfPublications;

    public Professor() {

    }

    public Professor(String name, int numberOfPublications) {

*super*(name);

*this*.numberOfPublications = numberOfPublications;

    }

    public int getNumberOfPublications() {

        return *this*.numberOfPublications;

    }

    public void setNumberOfPublications(int numberOfPublications) {

*this*.numberOfPublications = numberOfPublications;

    }

    @Override

    public boolean isOutstanding() {

        if (*this*.numberOfPublications >= 50) {

            return true;

        }

        return false;

    }

    @Override

    public String toString() {

        return *super*.toString() +  "{" +

            " numberOfPublications='" + getNumberOfPublications() + "'" +

            "}";

    }

}

## Student

public class Student extends Person {

    private double CGPA;

    public Student() {

    }

    public Student(String name, double CGPA) {

*super*(name);

*this*.CGPA = CGPA;

    }

    public double getCGPA() {

        return *this*.CGPA;

    }

    public void setCGPA(double CGPA) {

*this*.CGPA = CGPA;

    }

    @Override

    public boolean isOutstanding() {

        if (*this*.CGPA >= 3.5) {

            return true;

        }

        return false;

    }

    @Override

    public String toString() {

        return *super*.toString() +  "{" +

            " CGPA='" + getCGPA() + "'" +

            "}";

    }

}

## Runner

public class Runner {

    public static void main(String[] args) {

        Person fadwi[] = new Person[2];

        fadwi[0] = new Student("Shahzaneer Ahmed",3.23);

        fadwi[1] = new Professor("Molana Room", 52);

        System.out.println("Before Downcasting");

        for (int i = 0; i < fadwi.length; i++) {

            System.out.println(fadwi[i].toString());

            System.out.println(fadwi[i].isOutstanding());

*// downcasting*

            if (fadwi[i] instanceof Professor) {

                Professor p1 = (Professor) fadwi[i];

                p1.setNumberOfPublications(100);

                fadwi[i] = p1;

            }

        }

        System.out.println("After downcasting");

        for (int i = 0; i < fadwi.length; i++) {

            System.out.println(fadwi[i].toString());

            System.out.println(fadwi[i].isOutstanding());

        }

    }

}