Conor Peckham

CS 4051

Exercises O8-O16

**import** java.util.Scanner;

**public** **class** ExerciseO8 {

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

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

Employee emp1 = **new** Employee();

emp1.readPInfo(S);

emp1.readPayInfo(S);

System.***out***.println(emp1.getPInfoString());

System.***out***.println(emp1.getPayInfoString());

BonusEmployee emp2 = **new** BonusEmployee();

emp2.readPInfo(S);

emp2.readPayInfo(S);

System.***out***.println(emp2.getPInfoString());

System.***out***.println(emp2.getPayInfoString());

Employee emp3 = **new** Employee("John", "Doe", 111111, **new** Date(10,25,1991), **new** Date(05,10,2010), 1250);

System.***out***.println(emp3.getPInfoString());

System.***out***.println(emp3.getPayInfoString());

BonusEmployee emp4 = **new** BonusEmployee("Jobe", "Daly", 222222, **new** Date(01,05,1990), **new** Date (06,30,2011), 850, 250);

System.***out***.println(emp4.getPInfoString());

System.***out***.println(emp4.getPayInfoString());

}

}

**import** java.util.Scanner;

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

**public** Employee(){

IDNumber = 999999;

basePay = 0.00;

}

**public** Employee(String fname, String lname, **int** ID, Date bday, Date hire, **double** bPay)

{

firstName = fname;

lastName = lname;

IDNumber = ID;

Birthday = bday;

Hired = hire;

basePay = bPay;

}

**private** String firstName;

**private** String lastName;

**private** **int** IDNumber;

**private** Date Birthday = **new** Date();

**private** Date Hired = **new** Date();

**private** **double** basePay;

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

**void** readPInfo(Scanner scan) {

System.***out***.println("First Name?");

firstName = scan.nextLine();

System.***out***.println("Last Name?");

lastName = scan.nextLine();

System.***out***.println("ID number?");

IDNumber = scan.nextInt();

System.***out***.println("Birthday?");

Birthday.inputDate(scan);

System.***out***.println("Hired?");

Hired.inputDate(scan);

}

**void** readPayInfo(Scanner scan) {

System.***out***.println("Base pay?");

basePay = scan.nextDouble();

}

String getPInfoString() {

String info = "Name: " + lastName + ", " + firstName + "\n\n ID Number: " + IDNumber + "\n\n BIRTH DAY: " + Birthday.getStringDate() + "\n\n DATE HIRED: " + Hired.getStringDate() + "\n";

**return** info;

}

**void** setBpay(**double** newBpay) {

basePay = newBpay;

}

**double** getBpay() {

**return** basePay;

}

**double** getGpay() {

**return** basePay;

}

**double** computeTax() {

**double** taxed;

**if**(basePay >= 1000)

taxed = basePay \* .2;

**else** **if**(basePay >= 800 && basePay < 1000)

taxed = basePay \* .18;

**else** **if**(basePay >= 600 && basePay < 800)

taxed = basePay \* .15;

**else**

taxed = basePay \* .1;

**return** taxed;

}

String getPayInfoString() {

String info;

info = " GROSS PAY: " + basePay + "\n\n TAX DEDUCTION: " + computeTax() + "\n\n NET PAY: " + (basePay-computeTax());

**return** info;

}

}

**import** java.util.Scanner;

**public** **class** BonusEmployee **extends** Employee{

**private** **double** bonus;

**private** **double** basePay;

**public** BonusEmployee() {

bonus = 0.00;

}

**public** BonusEmployee(String fname, String lname, **int** iD, Date bday, Date hire, **double** basePay, **double** bon) {

**super**(fname, lname, iD, bday, hire, basePay);

bonus = bon;

}

@Override

**public** **void** readPayInfo(Scanner scan) {

System.***out***.print("\n Enter bonus: ");

basePay = scan.nextDouble();

System.***out***.print("\n Enter bonus: ");

bonus = scan.nextDouble();

}

@Override

**public** **double** getGpay()

{

**return**(basePay + bonus);

}

@Override

**public** String getPayInfoString()

{

**double** netpay = basePay - computeTax();

String payInfo = String.*format*("\nBase Pay: " + basePay + " \nBonus Pay: " + bonus + "\nGross Pay: " + getGpay() + "\nTax Deduction: " + computeTax() + "\nNet pay:" +netpay);

**return** payInfo;

}

}

**import** java.util.Scanner;

**public** **class** Date {

**public** Date() {

Month = 1;

Day = 1;

Year = 1960;

checkDate(Month, Day, Year);

}

**int** Month, Day, Year;

**public** Date(**int** Month, **int** Day, **int** Year)

{

Month = Month;

Day = Day;

Year = Year;

checkDate(Month, Day, Year);

}

**private** **void** checkDate(**int** Month, **int** Day, **int** Year)

{

**if**(Month <= 0 || Month >= 13) {

System.*exit*(1);

}

**else** **if**(Month == 9 || Month == 4 || Month == 6 || Month == 11)

{

**if**(Day <= 0 || Day >= 31)

{

System.*exit*(1);

}

}

**else** **if**(Month == 2)

{

**if**(Day <= 0 || Day >= 29) {

System.*exit*(1);

}

}

**else** **if**(Year < 1960 || Year > 2016)

{

System.*exit*(1);

}

}

**public** **void** inputDate(Scanner scan)

{

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

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

Month = scan.nextInt();

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

Day = scan.nextInt();

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

Year = scan.nextInt();

}

**public** String getStringDate()

{

String month = "" + Month;

String day = "" + Day;

String year = "" + Year;

String date = month + "/" + day + "/" + year;

**return** date;

}

**public** **boolean** isEqualTo(Date obj)

{

**if**(Month == obj.Month && Day == obj.Day && Year == obj.Year)

**return** **true**;

**else**

**return** **false**;

}

**public** **int** getMonth()

{

**return** Month;

}

**public** **int** getDay()

{

**return** Day;

}

**public** **int** getYear()

{

**return** Year;

}

**public** **boolean** isGreaterThan(Date obj1, Date obj2)

{

**if**(obj1.Year >= obj2.Year && obj1.Month>= obj2.Month && obj1.Day > obj2.Day)

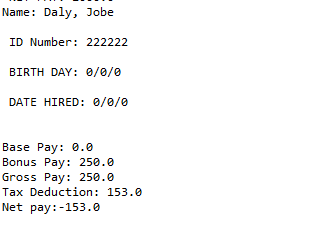
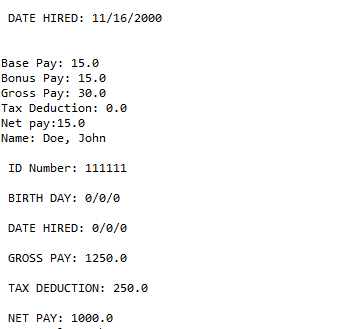
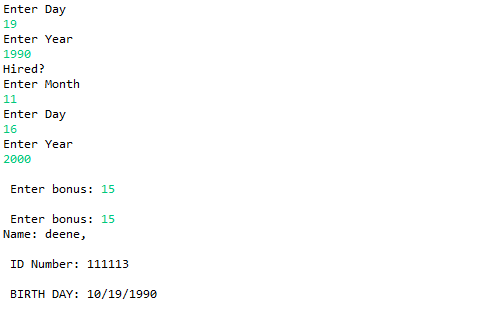
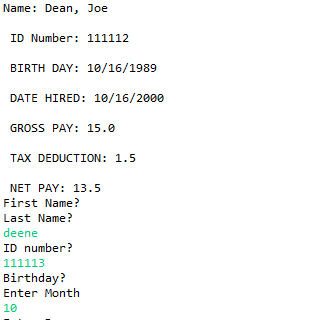
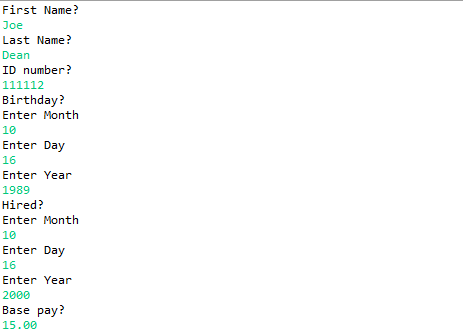
**return** **true**;

**else**

**return** **false**;

}

}



**import** java.util.Scanner;

**public** **class** ExerciseO9 {

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

{

Employee eemp = **new** Employee( );

Employee bemp1 = **new** BonusEmployee( );

BonusEmployee bemp2 = **new** BonusEmployee( );

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

eemp.readPayInfo(S);

bemp1.readPayInfo(S);

bemp2.readPayInfo(S);

System.***out***.println( "Gross Pay of eemp=\t" + (eemp.getGpay()));

System.***out***.println( "Gross Pay of bemp1=\t" + (bemp1.getGpay()));

System.***out***.println( "Gross Pay of bemp2=\t" + (bemp2.getGpay()));

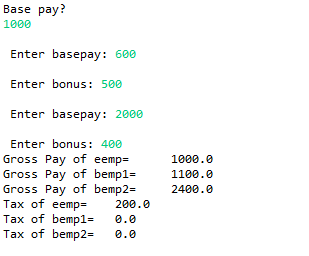
System.***out***.println( "Tax of eemp=\t" + (eemp.computeTax()));

System.***out***.println( "Tax of bemp1=\t" + (bemp1. computeTax ()));

System.***out***.println( "Tax of bemp2=\t" + (bemp2. computeTax ()));

}

}



**public** **class** ExerciseO11 {

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

{

Date d = **new** Date();

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

BonusEmployee be = **new** BonusEmployee();

Date today = **new** Date();

System.***out***.println("Date: " + d);

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

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

System.***out***.println(today.getClass().getName());

}

}

**import** java.util.Scanner;

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

**public** Employee(){

IDNumber = 999999;

basePay = 0.00;

}

**public** Employee(String fname, String lname, **int** ID, Date bday, Date hire, **double** bPay)

{

firstName = fname;

lastName = lname;

IDNumber = ID;

Birthday = bday;

Hired = hire;

basePay = bPay;

}

**protected** String firstName;

**protected** String lastName;

**protected** **int** IDNumber;

**protected** Date Birthday = **new** Date();

**protected** Date Hired = **new** Date();

**private** **double** basePay;

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

**void** readPInfo(Scanner scan) {

System.***out***.println("First Name?");

firstName = scan.nextLine();

System.***out***.println("Last Name?");

lastName = scan.nextLine();

System.***out***.println("ID number?");

IDNumber = scan.nextInt();

System.***out***.println("Birthday?");

Birthday.inputDate(scan);

System.***out***.println("Hired?");

Hired.inputDate(scan);

}

**void** readPayInfo(Scanner scan) {

System.***out***.println("Base pay?");

basePay = scan.nextDouble();

}

String getPInfoString() {

String info = "Name: " + lastName + ", " + firstName + "\n\n ID Number: " + IDNumber + "\n\n BIRTH DAY: " + Birthday.getStringDate() + "\n\n DATE HIRED: " + Hired.getStringDate() + "\n";

**return** info;

}

**void** setBpay(**double** newBpay) {

basePay = newBpay;

}

**double** getBpay() {

**return** basePay;

}

**double** getGpay() {

**return** basePay;

}

**double** computeTax() {

**double** taxed;

**if**(basePay >= 1000)

taxed = basePay \* .2;

**else** **if**(basePay >= 800 && basePay < 1000)

taxed = basePay \* .18;

**else** **if**(basePay >= 600 && basePay < 800)

taxed = basePay \* .15;

**else**

taxed = basePay \* .1;

**return** taxed;

}

String getPayInfoString() {

String info;

info = " GROSS PAY: " + basePay + "\n\n TAX DEDUCTION: " + computeTax() + "\n\n NET PAY: " + (basePay-computeTax());

**return** info;

}

@Override

**public** String toString()

{

String s = String.*format*("Name: " + lastName + ", " + firstName + "ID Number: " + IDNumber + "Birthday: " + Birthday.getStringDate() + " Date Hired: " + Hired.getStringDate() + "Gross Pay: " + getGpay() + " Tax Deduction: " + computeTax());

**return** s;

}

}

**import** java.util.Scanner;

**public** **class** BonusEmployee **extends** Employee{

**private** **double** bonus;

**private** **double** basePay;

**public** BonusEmployee() {

bonus = 0.00;

}

**public** BonusEmployee(String fname, String lname, **int** iD, Date bday, Date hire, **double** basePay, **double** bon) {

**super**(fname, lname, iD, bday, hire, basePay);

bonus = bon;

}

@Override

**public** **void** readPayInfo(Scanner scan) {

System.***out***.print("\n Enter bonus: ");

basePay = scan.nextDouble();

System.***out***.print("\n Enter bonus: ");

bonus = scan.nextDouble();

}

@Override

**public** **double** getGpay()

{

**return**(basePay + bonus);

}

@Override

**public** String getPayInfoString()

{

**double** netpay = basePay - computeTax();

String payInfo = String.*format*("\nBase Pay: " + basePay + " \nBonus Pay: " + bonus + "\nGross Pay: " + getGpay() + "\nTax Deduction: " + computeTax() + "\nNet pay:" +netpay);

**return** payInfo;

}

@Override

**public** String toString()

{

String s = String.*format*("Name: " + lastName + ", " + firstName + "ID Number: " + IDNumber + "Birthday: " + Birthday.getStringDate() + " Date Hired: " + Hired.getStringDate() + "Gross Pay: " + getGpay() + " Tax Deduction: " + computeTax());

**return** s;

}

}

import java.util.Scanner;

public class Date {

public Date() {

Month = 1;

Day = 1;

Year = 1960;

checkDate(Month, Day, Year);

}

int Month, Day, Year;

private void checkDate(int Month, int Day, int Year)

{

if(Month <= 0 || Month >= 13) {

System.exit(0);

}

else if(Month == 9 || Month == 4 || Month == 6 || Month == 11)

{

if(Day <= 0 || Day >= 31)

{

System.exit(0);

}

}

else if(Month == 2)

{

if(Day <= 0 || Day >= 29) {

System.exit(0);

}

}

else if(Year < 1960 || Year > 2016)

{

System.exit(0);

}

}

public void inputDate(Scanner scan)

{

scan = new Scanner(System.in);

System.out.println("Enter Month");

Month = scan.nextInt();

System.out.println("Enter Day");

Day = scan.nextInt();

System.out.println("Enter Year");

Year = scan.nextInt();

}

public String getStringDate()

{

String month = "" + Month;

String day = "" + Day;

String year = "" + Year;

String date = month + "/" + day + "/" + year;

return date;

}

public boolean isEqualTo(Date obj)

{

if(Month == obj.Month && Day == obj.Day && Year == obj.Year)

return true;

else

return false;

}

public int getMonth()

{

return Month;

}

public int getDay()

{

return Day;

}

public int getYear()

{

return Year;

}

public boolean isGreaterThan(Date obj1, Date obj2)

{

if(obj1.Year >= obj2.Year && obj1.Month>= obj2.Month && obj1.Day > obj2.Day)

return true;

else

return false;

}

@Override

public String toString()

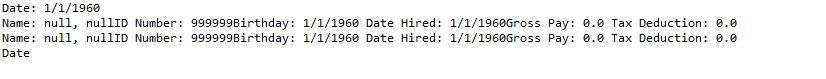
{

String s = getStringDate();

return s;

}

}



**public** **class** ExerciseO12 {

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

{

Tile tile1 = **new** Rectangle(3.0, 5.0, 4.0);

Tile tile2 = **new** Triangle(2.0, 6.0, 8.0);

tile1.print();

tile2.print();

}

}

**public** **abstract** **class** Tile {

**private** **double** unitPrice;

**public** Tile( )

{

unitPrice = 2.0;

}

**public** Tile (**double** uprice)

{

unitPrice = uprice;

}

**public** **double** getUprice()

{

**return** unitPrice ;

}

**public** **abstract** **double** computeArea(); // abstract method

**public** **double** computePrice()

{

**double** area;

area = computeArea(); // calling the abstract method

**return**(unitPrice \* area);

}

**public** **void** print()

{

System.***out***.println("The price of the tile is:\t" + computePrice ());

}

}

**public** **class** Rectangle **extends** Tile // inherits class Tile

{

**private** **double** length;

**private** **double** width;

**public** Rectangle()

{

length = 0.5;

width = 1.0;

}

**public** Rectangle(**double** len, **double** wth, **double** uprice)

{

**super**(uprice);

length = len;

width = wth;

}

@Override

**public** **double** computeArea()

{

**return** (length \* width); // return the area of a rectangular tile

}

@Override

**public** **void** print()

{

System.***out***.println("The length is:\t" + length);

System.***out***.println("The width is:\t" + width);

System.***out***.println("The price of the tile is:\t" + computePrice());

}

}

**public** **class** Triangle **extends** Tile // inherits class Tile

{

**private** **double** height;

**private** **double** base;

**public** Triangle()

{

height = 0.5;

base = 1.0;

}

**public** Triangle(**double** ht, **double** bse, **double** uprice)

{

**super**(uprice);

height = ht;

base = bse;

}

@Override

**public** **double** computeArea()

{

**return** ( height \* base/2.0 ); // return the area of a triangular tile

}

@Override

**public** **void** print()

{

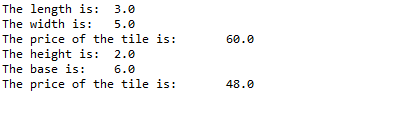
System.***out***.println( "The height is:\t" + height);

System.***out***.println( "The base is:\t" + base);

System.***out***.println( "The price of the tile is:\t" + computePrice ( ));

}

}



**public** **class** ExerciseO13 {

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

{

HourlyPay Pay1 = **new** HourlyPay();

Pay1.printDetails();

HourlyPay Pay2 = **new** HourlyPay(50, 12);

Pay2.printDetails();

}

}

**public** **interface** Payroll {

**double** ***TAXRATE*** = 0.15;

**public** **abstract** **double** getGrossPay();

**double** getDeduction();

**void** printDetails();

}

**class** HourlyPay **implements** Payroll

{

**private** **int** hours;

**private** **double** payRate;

**public** HourlyPay()

{

hours = 40;

payRate = 10.00;

}

**public** HourlyPay(**int** hour, **double** pay)

{

hours = hour;

payRate = pay;

}

**private** **double** getOverTime()

{

**if**(hours > 40)

{

**return** (1.5 \* (hours - 40) \* payRate);

}

**else**

{

**return** 0;

}

}

@Override

**public** **double** getGrossPay()

{

**return** (hours \* payRate + getOverTime());

}

@Override

**public** **double** getDeduction()

{

**return** (***TAXRATE*** \* getGrossPay());

}

@Override

**public** String toString()

{

String s = String.*format*("\nPAY RATE:\t" + payRate + "\nHOURS:\t" + hours + "\nOVERTIME:\t" + getOverTime() + "\nGROSS PAY:\t" + getGrossPay() + "\nTAX DEDUCTION:\t" + getDeduction() + "\nNET PAY:\t" + (getGrossPay() - getDeduction()));

**return** s;

}

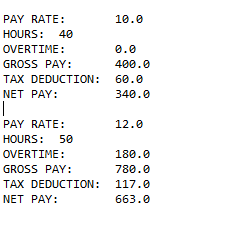
**public** **void** printDetails()

{

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

}

}



**public** **class** ExerciseO14 {

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

{

/\*--- 1 ------ write the sequence of statements to call the method outerMethod ------------\*/

OuterClass outer = **new** OuterClass();

outer.outerMethod();

/\*--- 2 ------ write the sequence of statements to call the method innerMethod2 ------------\*/

OuterClass.NestedClass nested = **new** OuterClass.NestedClass();

nested.innerMethod2();

}

}

**class** OuterClass

{

**public** **void** outerMethod( )

{

System.***out***.println("outer class");

/\*---- 3 ----- write the sequence of statements to call the method innerMethod1 ------------\*/

NestedClass nest = **new** NestedClass();

nest.innerMethod1();

}

**static** **class** NestedClass

{

**public** **void** innerMethod1( )

{

System.***out***.println("nested class Method1");

}

**public** **void** innerMethod2( )

{

System.***out***.println("nested class Method2");

/\*---- 4 ----- write the sequence of statements to call the method outerMethod------------\*/

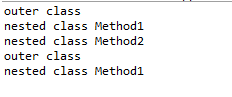
OuterClass obj = **new** OuterClass();

obj.outerMethod();

}

} // end of class StaticNestedClass

} // end of class OuterClass



**public** **class** ExerciseO15 {

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

{

/\*--- 1 ------ write the sequence of statements to call the method outerMethod -------------\*/

OuterClass outer = **new** OuterClass();

outer.outerMethod();

/\*--- 2 ------ write the sequence of statements to call the method innerMethod2 ------------\*/

OuterClass.InnerClass inner = outer.**new** InnerClass();

inner.innerMethod2();

}

}

**class** OuterClass

{

**private** **int** outerNum;

**public** OuterClass( )

{

outerNum = 20;

}

**public** **int** getOuterNum( )

{

**return**( outerNum );

}

**public** **void** outerMethod( )

{

System.***out***.println("outer class");

/\*---- 3 ----- write the sequence of statements to call the method innerMethod1 ------------\*/

OuterClass.InnerClass inside = **new** OuterClass.InnerClass();

inside.innerMethod1();

/\*-- 4 -- write the sequence of statements to output the value of instance variable innerNum -\*/

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

}

**class** InnerClass

{

**private** **int** innerNum;

**public** InnerClass( )

{

innerNum = 15;

}

**public** **int** getInnerNum( )

{

**return**( innerNum );

}

**public** **void** innerMethod1( )

{

System.***out***.println("nested class 1");

}

**public** **void** innerMethod2( )

{

System.***out***.println("nested class 2");

/\*---- 5 ----- write the sequence of statements to call the method outerMethod------------\*/

OuterClass outside = **new** OuterClass();

outside.outerMethod();

/\* 6 - write the sequence of statements to output the value of instance variable outerNum -\*/

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

}

} // end of class InnerClass

} // end of class OuterClass

