**Exam 1 Instructions**

**OBJECT-ORIENTED PROG**

**FALL 2020**

* This is a take-home exam. You can use any resources that are available for you to finish this exam, ***except***
  + Outsourcing the exam to any person or any third-party websites
  + Copying from other students work
  + Copying direct quotes from the books or internet
* Do not lose your opportunity to learn while working on the exam. Understand the concept and write answers on your own.
* Usually, in life, we have several choices. Unfortunately, you don’t have any choice on this exam. You have to answer all the questions and each part of the problem.
* All the topics on this exam were discussed in class before week 6. So, you cannot claim that the questions are out of the syllabus!
* Refer to Microsoft Word tutorials for proper formatting
* Points will be deducted for grammatical, spelling mistakes and missing source code comments
* No two brains think alike unless you are soulmates. Definitely, your answers will not be the same as other students.
* If you have any doubts about the exam, please answer it based on your assumptions and mention them as a part of your answers.
* Organize your source code for every question in a respective package. Push your source code to GitHub and provide your GitHub link at the end of the document.
* Read the code of academic integrity before you start the exam. <https://www.nwmissouri.edu/policies/academics/Academic-Integrity.pdf>

**Exam 1 OBJECT-ORIENTED PROG 01FA20 100 pts**

1. (5 Points) What are syntax errors, run time errors and logic errors? Explain and demonstrate with examples.

**Syntax Errors:** The errors that occur during the compilation time are known as Syntax errors. When this error occurs, it is not possible to execute the program. For example: If there is no semi-colon or syntax is wrong. Let us consider that we write a line in a code like- int num and proceed then there will be an syntax error that a semi colon has to be declared, i.e., int num; Also if a variable is used without declaring- number=25; then an error occurs that says that variable has not been declared.   
  
  
  
Graphical user interface, text, application, email

Description automatically generated  
  
  
Graphical user interface, text, application

Description automatically generated

**Run Time errors:** These errors usually occur in the running time of the program. Lets us consider we declare a variable which is int number; later in the run time you give an alphabet or name which is not a number, then an exception is shown which is nothing but a run time error.  
  
Graphical user interface, text, application

Description automatically generated

**Logic Errors:** A logic error occurs when there is something wrong in the program even though it has been compiled and executed. These errors can be identified if there is error in logic. It is not easy to find out the logic error unless until we know where the mistake has been done. For example : Division with zero.  
  
Graphical user interface, text, application, email

Description automatically generated

1. (5 Points) Design a class named Fan to represent a fan. The class contains:
2. Three constants named **SLOW, MEDIUM, a**nd **FAST** with the values **1, 2,** and **3** to denote the fan speed.
3. A private **int** data field named **speed** that specifies the speed of the fan
4. A private **boolean** data field named **on** that specifies whether the fan is on
5. A private **double** data field named **radius** that specifies the radius of the fan
6. A **string** data field named **color** that specifies the color of the fan
7. The accessor and mutator methods for all four data fields.
8. A no-arg constructor that creates a default fan.
9. A method named **toString()** that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and radius along with the string “fan is off” in one combined string.
10. Write a Driver class to test all the methods of Fan class and provide sample input and output

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question2;

/\*\*

\*

\* @author Sai Prashansa

\*/

public class Fan {

public static final int SLOW= 1;

public static final int MEDIUM=2;

public static final int FAST= 3;

private int speed;

private boolean on;

private double radius;

String color;

//no-arg constructor

public Fan() {

}

//getters and setters

public int getSpeed() {

return speed;

}

public boolean isOn() {

return on;

}

public double getRadius() {

return radius;

}

public String getColor() {

return color;

}

public void setSpeed(int speed) {

this.speed = speed;

}

public void setOn(boolean on) {

this.on = on;

}

public void setRadius(double radius) {

this.radius = radius;

}

public void setColor(String color) {

this.color = color;

}

//to-string that returns fan description

@Override

public String toString() {

if(on==true && speed==1){

return "Speed=" + Fan.SLOW + "(Slow), Radius=" + radius + ", Color=" + color;

}

else if(on==true && speed==2){

return "Speed=" + Fan.MEDIUM + "(Medium), Radius=" + radius + ", Color=" + color;

}

else if(on==true && speed==3){

return "Speed=" + Fan.FAST + "(Fast), Radius=" + radius + ", Color=" + color;

}

else{

return "Fan is off. "+"Radius=" + radius + ", Color=" + color;

}

}

}

//Driver Class

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question2;

import java.util.Scanner;

/\*\*

\*

\* @author Sai Prashansa

\*/

public class FanDriver {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

String color;

double radius;

int speed;

boolean on;

Fan f1=new Fan(); //object creation

Scanner scan = new Scanner(System.in);

System.out.print("Enter Fan Color:");

color = scan.nextLine();

//validation for color

while(color.equals("") || color.equals(null)){

System.out.print("Please enter fan color: ");

color = scan.nextLine();

}

f1.setColor(color);

System.out.print("Enter Fan Radius:");

radius = scan.nextDouble();

//validation for radius

while(radius<=0){

System.out.print("Radius cannot be zero. Please enter valid radius: ");

radius = scan.nextDouble();

}

f1.setRadius(radius);

System.out.print("Is fan on?(true/false): ");

on= scan.nextBoolean();

f1.setOn(on);

if(on==true){

System.out.print("Enter fan speed: ");

speed= scan.nextInt();

while(speed>=4 || speed<=0){

System.out.print("Please enter values 1,2 or 3: ");

speed= scan.nextInt();

}

f1.setSpeed(speed);

}

System.out.println("\n\*\*\*\*\*Getter Method\*\*\*\*\*");

System.out.println("Radius: " + f1.getRadius());

System.out.println("Color: " + f1.getColor());

if(f1.isOn()==true){

System.out.println("Fan status: ON");

System.out.println("Speed: "+f1.getSpeed());

}else{

System.out.println("Fan Status: OFF");

}

System.out.println("\n\*\*\*\*\*ToString Method\*\*\*\*\*\n"+f1.toString());

}

}

Output:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. (5 Points) Write the output for below code and explain the concept of equality comparison? Use figures to explain the references.

|  |
| --- |
| Rectangle box1 = new Rectangle(10,20,10,10);  Rectangle box2 = new Rectangle(10,20,10,10);  Rectangle box3 = box1;  Rectangle box4;  System.out.println(box1 == box2);  System.out.println(box1.equals(box2));  System.out.println(box2 == box3);  System.out.println(box2.equals(box3));  System.out.println(box1==box3);  box3 = new Rectangle(10,20,10,10);  System.out.println(box1==box3);  System.out.println(box1.equals(box3));  String s1 = new String(“Java”);  String s2 = s1;  String s3 = new String(s2);  String s4 = “Java”;  System.out.println(s1 == s2);  System.out.println(s1.equals(s2));  System.out.println(s1 == s3);  System.out.println(s1.equals(s3));  System.out.println(s1 == s4);  System.out.println(s1.equals(s4));  System.out.println(s2 == s3);  System.out.println(s2.equals(s3));  System.out.println(s2 == s4);  System.out.println(s2.equals(s4));  System.out.println(s3 == s4);  System.out.println(s3.equals(s4)); |

Output:  
  
Graphical user interface, text, application

Description automatically generated  
  
false  
true  
false  
true  
true  
false  
true  
true  
true  
false  
true  
false  
true  
false  
true  
false  
true  
false  
true

**Equality Comparison:**In java we use “equals()” method to check the content in the object. Whereas, “==” is an operator which is used to compare the address of both the objects.   
  
In the given code  
  
Line 1: A user defined class “Rectangle” with object “box1” is created which takes four parameters and is initialized in a new memory location.   
Line 2: Similar to line 1, another “rectangle” object “box2” is initialized in another new memory location using key word new.  
Line 3: Here a “box3” is created and “box1” is assigned to it.  
Line 4: A “box4” object is created and not initialized. So, no memory is allocated.   
  
Hence wherever there has been “==” comparison the address has been checked and returned if it was true/false. Similarly whenever there was “.equals()” it checked the content in the object and returned if it was true/false.

Rectangle

10,20,10,10

Box1

Box3

Rectangle

10,20,10,10

Box2

Box4

String

Java

S1

S2

String

S2

S3

S4

Java

1. (5 Points) Write a program that prompts the user to enter a Social Security number in the format DDD-DD-DDDD, where D is a digit. Your program should check whether the input is valid. Provide sample input and output.

Here is a sample run:

|  |
| --- |
| Enter a SSN: 232-23-5435  232-23-5435 is a valid social security number |

|  |
| --- |
| Enter a SSN: 23-23-5435  23-23-5435 is an invalid social security number |

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question4;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class SocialSecurity {

String ssn;

oolean dash;

oolean length;

//public method of return type String to check the right format, with paramters ssn of return type String

public String CorrectFormat(String ssn){

if(ssn.length()== 11){

this.length=true;

if(length==true){

if((ssn.charAt(3) == ‘-‘)&&(ssn.charAt(6) == ‘-‘)){ //checks if the character at 3,6 is a dash(-)

this.dash=true;

}

else{

return ssn + “ is a invalid social security number”;

}

oolean digit0= Character.isDigit(ssn.charAt(0));

oolean digit1= Character.isDigit(ssn.charAt(1));

oolean digit2= Character.isDigit(ssn.charAt(2));

oolean digit3= Character.isDigit(ssn.charAt(4));

oolean digit4= Character.isDigit(ssn.charAt(5));

oolean digit5= Character.isDigit(ssn.charAt(7));

oolean digit6= Character.isDigit(ssn.charAt(8));

oolean digit7= Character.isDigit(ssn.charAt(9));

oolean digit8= Character.isDigit(ssn.charAt(10));

//if loop to check if the characters at 0,1,2,4,5,7,8,9,10 are digits

if(digit1==true && digit2== true && digit3==true && digit4==true

&& digit5==true && digit6==true && digit7==true && digit8==true

&& digit0==true){

return ssn + “ is a valid social security number”;

}

else{

return ssn + “ is an invalid social security number”;

}

}

}

else{

return ssn + “ is a invalid social security number”;

}

return “Enter valid ssn”;

}

}

//DriverClass

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question4;

import java.util.Scanner;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class SocialSecurityDriver {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

String ssn;

System.out.print(“Enter a Social Security Number: “);

Scanner scan= new Scanner(System.in);

ssn= scan.nextLine();

SocialSecurity ss = new SocialSecurity();

System.out.println(ss.CorrectFormat(ssn));

}

}

Output:

Graphical user interface, application, Word

Description automatically generated  
  
Graphical user interface, application, Word

Description automatically generated

1. (No Mistakes)
2. (No Mistakes)

1. (5 Points) The great circle distance is the distance between two points on the surface of a sphere. Let (x1, y1) and (x2, y2) be the geographical latitude and longitude of two points. The great circle distance between the two points can be computed using the following formula:

Write a program that prompts the user to enter the latitude and longitude of two points on the earth in degrees and displays its great circle distance. The average earth radius is 6,371.01 km. Note that you need to convert the degrees into radians using the **Math.toRadians** method since the Java trigonometric methods use radians. The latitude and longitude degrees in the formula are for north and west. Use negative to indicate south and east degrees. After executing the program, what did you notice with Radians and degrees? Here is a sample run:

|  |
| --- |
| Enter point 1 (latitude and longitude) in degrees: 39.55, -116.25  Enter point 2 (latitude and longitude) in degrees: 41.5, 87.37  The distance between the two points is 10691.79183231593 km |

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question7;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class LatitudeAndLongitude {

double x1,x2,y1,y2;

double d;

double radius=6371.01;

//a public double method to find the distance between two points of the great circle

public double distance(double x1, double x2, double y1, double y2){

x1=Math.toRadians(x1);

x2=Math.toRadians(x2);

y1=Math.toRadians(y1);

y2=Math.toRadians(y2);

double eq = Math.sin(x1)\*Math.sin(x2)+

Math.cos(x1)\*Math.cos(x2)\*

Math.cos(y1-y2);

d= radius \* Math.acos(eq);

return d;

}

}

//DriverClass

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question7;

import java.util.Scanner;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class LatitudeAndLongitudeDriver {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Scanner scan=new Scanner(System.in);

System.out.print("Enter point1 (latitude and longitude) in degrees: ");

String lat = scan.nextLine();

int delim = 0;

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

if(lat.charAt(i) == ','){

delim = i;

}

}

double x1 = Double.parseDouble(lat.substring(0, delim));

double y1 = Double.parseDouble(lat.substring(delim+1,lat.length()));

System.out.print("Enter point2 (latitude and longitude) in degrees: ");

String lon = scan.nextLine();

int delim1 = 0;

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

if(lon.charAt(i) == ','){

delim1 = i;

}

}

double x2 = Double.parseDouble(lon.substring(0, delim1));

double y2 = Double.parseDouble(lon.substring(delim1+1,lon.length()));

LatitudeAndLongitude lal=new LatitudeAndLongitude();

double dist=lal.distance(x1, x2, y1, y2);

System.out.println("The distance between the two points is "+dist+" km");

scan.close();

}

}

Output:

Graphical user interface, application, Word

Description automatically generated  
  
  
A radian is a unit measurement for angles defined by ratio of the length of the arc of a circle to the radius of the circle. Converting an angle measured in degrees to an equivalent angle measured in radians is done by using Math.toRadians(). Lets say we have given a degrees of 90.0 and want to convert it into radians, we use the Math.toRadians() and we can see an output of 1.570796, i.e., 1 degree = 0.0174 radians, (it is calculated by degree\*(PI/180)=radians).

1. (7 Points) Questions on Math class
   1. Evaluate the following method calls.

(a) Math.sqrt(4)

(b) Math.sin(2 \* Math.PI)

(c) Math.cos(2 \* Math.PI)

(d) Math.pow(2, 2)

(e) Math.log(Math.E)

(f) Math.exp(1)

(g) Math.max(2, Math.min(3, 4))

(h) Math.rint(-2.5)

(i) Math.ceil(-2.5)

(j) Math.floor(-2.5)

(k) Math.round(-2.5f)

(l) Math.round(-2.5)

(m) Math.rint(2.5)

(n) Math.ceil(2.5)

(o) Math.floor(2.5)

(p) Math.round(2.5f)

(q) Math.round(2.5)

(r) Math.round(Math.abs(-2.5))

**OUTPUT:**  
  
Graphical user interface, application, Word

Description automatically generated  
  
2.0  
  
-2.4492935982947064E-16  
  
1.0  
  
4.0  
  
1.0  
  
2.718281828459045  
  
3  
  
-2.0  
  
-2.0  
  
-3.0  
  
-2  
  
-2  
  
2.0  
  
3.0  
  
2.0  
  
3  
  
3  
  
3

* 1. True or false? The argument for trigonometric methods is an angle in radians.

True

* 1. Write a statement that converts **47** degrees to radians and assigns the result to a variable.

double rad = Math.toRadians(47);

System.out.println(rad);

* 1. Write a statement that converts **π/7** to an angle in degrees and assigns the result to a variable.

double pi=Math.toDegrees(Math.PI/7);  
System.out.println(pi);

* 1. Write an expression that obtains a random integer between **34** and **55.** Write an expression that obtains a random integer between **0** and **999.** Write an expression that obtains a random number between **5.5** and **55.5.**

//obtains random integer between 34 and 55

int num1=34 + (int)(Math.random()\*(55-34));

System.out.println("\n"+num1);

//obtains random integer between 0 to 999

int num2=(int)(Math.random() \* 1000);

System.out.println(num2);

//obtains random integer between 5.5 to 55.5

int num3= (int) (5.5 + (Math.random() \* (55.5 - 5.5)));

System.out.println(num3);  
  
Output:

Graphical user interface, application, Word

Description automatically generated

* 1. Why does the **Math** class not need to be imported?

When we have java.lang package imported, there will be no need of a math class as it is already in the java.lang package.

* 1. What is **Math.log(Math.exp(5.5))**? What is **Math.exp(Math.log(5.5))**?

What is **Math.asin(Math.sin(Math.PI / 6))**?

What is **Math.sin(Math.asin(Math.PI / 6))**?

5.5

5.5

0.5235987755982988

0.5235987755982988

1. (8 Points) Java API has the **GregorianCalendar** class in the **java.util** package, which you can use to obtain the year, month, and day of a date. The no-arg constructor constructs an instance for the current date, and the methods **get(GregorianCalendar.YEAR)**, get**(GregorianCalendar.MONTH),**and **get(GregorianCalendar.DAY\_OF\_MONTH)** return the year, month, and day.

Write a program to perform two tasks:

* Display the current year, month, and day.
* The **GregorianCalendar** class has the **setTimeInMillis(long),** which can be used to set a specified elapsed time since January 1, 1970. Set the value to **1234567898765L** and display the year, month, and day.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question9;

//GregorianCalender has been imported

import java.util.GregorianCalendar;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class Calendar {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

//no-arg constructor

GregorianCalendar gcal = new GregorianCalendar();

int year= gcal.get(GregorianCalendar.YEAR); //gets the current year

int month= gcal.get(GregorianCalendar.MONTH)+1; // gets the current month - 1. Hence we have to increment by 1

int day= gcal.get(GregorianCalendar.DAY\_OF\_MONTH); //gets the current day of the month

System.out.println("The current year is: "+year);

System.out.println("The current month is: "+month);

System.out.println("The current day of the month is: "+day);

System.out.println("Date:"+month+"-"+day+"-"+year);

//Set a specified elapsed time since January 1, 1970.Value=1234567898765L

gcal.setTimeInMillis(1234567898765L);

int yeare=gcal.get(GregorianCalendar.YEAR);

int monthe= gcal.get(GregorianCalendar.MONTH);

int daye= gcal.get(GregorianCalendar.DAY\_OF\_MONTH);

System.out.println("The current year is: "+yeare);

System.out.println("The current month is: "+monthe);

System.out.println("The current day of the month is: "+daye);

System.out.println("Date after value has been set time in millis:"+monthe+"-"+daye+"-"+yeare);

}

}

Output:

Graphical user interface, application, Word

Description automatically generated

* Write briefly what did you learn and observe from this program.

I could observe that when the java.util.package has been imported there has been no need to create any methods as it has a lot of methods inbuilt in it. This helps us to simply use the get methods and run the program. The GregorianCalendar class has methods to get Year, Month and Day. It just needs us to write a code to print the current date which can be done with simple get(GregorianCalender.Year) for year, get(GregorianCalender.month) for month and (GregorianCalender.day\_of\_month) for day of that month.

The month in GregorianCalender will take from 0 to 11, therefore we need to increment the month by 1.

1. (No Mistakes)
2. (10 Points) The monthly payment for a given loan pays the principal and the interest. The monthly interest is computed by multiplying the monthly interest rate and the balance (the remaining principal).The principal paid for the month is therefore the monthly payment minus the monthly interest. Write a program that lets the user enter the loan amount, number of years, and interest rate and displays the amortization schedule for the loan.

Here is a sample run:

|  |
| --- |
| Loan Amount: 10000  Number of Years: 1  Annual Interest Rate: 7  Monthly Payment: 865.26  Total Payment: 10383.21  Payment# Interest Principal Balance  1 58.33 806.93 9193.07  2 53.62 811.64 8381.43  ...  11 10.0 855.26 860.27  12 5.01 860.25 0.01 |

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question11;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class Loan {

double loanAmount;

double years;

double interestRate;

double monthlyIR;

double principleAmount;

double monthlypay;

double monthlyrate;

double monthlyamount;

double balance;

private double totalPayment=0;

//Constructor

public Loan(double loanAmount, double years, double interestRate) {

this.loanAmount = loanAmount;

this.years = years;

this.interestRate = interestRate;

this.balance = loanAmount;

}

//public method to get monthly interest

public double monthlyInt(){

monthlyIR = (interestRate/100)/ 12;

return monthlyIR;

}

//public double method to calculate the monthly pay

public double monthlyPay(){

double monthlyamount = balance \* monthlyInt() / (1 - (1 /Math.pow(1 + monthlyInt(), years \* 12)));

return monthlyamount;

}

//public method to calculate total pay

public double totalPay(){

totalPayment = (monthlyPay()\*12)\*years;

return totalPayment;

}

//public method to get interest

public double getMonthlyIntPay(){

double monthlyrate=loanAmount\*monthlyIR;

return monthlyrate;

}

//public method to get principle amount

public double getprincipleAmount(){

principleAmount = monthlyPay()-getMonthlyIntPay();

return principleAmount;

}

//public method to get balance

public double getbalance(){

loanAmount = loanAmount - getprincipleAmount();

return loanAmount;

}

}

//DriverClass

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question11;

import java.util.Scanner;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class LoanDriver {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Scanner scan=new Scanner(System.in);

System.out.print("Loan Amount: ");

double loanAmount = scan.nextDouble();

System.out.print("Number of Years: ");

int years = scan.nextInt();

System.out.print("Annual Interest Rate: ");

double interestRate = scan.nextDouble();

Loan l=new Loan(loanAmount,years,interestRate);

System.out.printf("\nMonthly payment: %.2f",l.monthlyPay());

System.out.printf("\nTotal payment: %.2f",l.totalPay());

System.out.println("\n\nPayment# Interest Principal Balance");

for(int i=1;i<=years\*12;i++){

System.out.printf("%-13d%-13.2f%-13.2f%-15.2f\n",i,l.getMonthlyIntPay(),l.getprincipleAmount(),l.getbalance());

}

}

}

/\*

Output:

Graphical user interface, application, Word

Description automatically generated

1. (10 Points) You have just started a sales job in a department store. Your pay consists of a base salary and a commission. The base salary is $5,000. The scheme shown below is used to determine the commission rate.

|  |  |
| --- | --- |
| **Sales Amount** | **Commission Rate** |
| $0.01-$5,000 | 8% |
| $5,000.01-$10,000 | 10% |
| $10,000.01 and above | 12% |

Note that this is a graduated rate. The rate for the first $5,000 is at 8%, the next $5000 is at 10%, and the rest is at 12%. If the sales amount is 25,000, the commission is 5,000 \* 8% + 5,000 \* 10% + 15,000 \* 12% = 2,700.

Your goal is to earn $30,000 a year. Write a program that finds the minimum sales you have to generate in order to make $30,000. Use do-while loop.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package question12;

/\*\*

\*

\* @author Sai Prashansa Ambarkar

\*/

public class SalesJobDriver {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

double goal=30000;

double commission=0;

double salesAmount = 0;

double amountbal=0;

do{

amountbal =0;

commission = 0;

if(salesAmount > 10000){

amountbal = salesAmount - 10000;

commission = commission + (amountbal \* 0.12);

}

if(salesAmount > 5000){

amountbal = amountbal - (amountbal - 5000);

commission = commission + (amountbal \* 0.10);

}

if(salesAmount > 0) {

commission=commission+ (amountbal\*0.08);

}

salesAmount = salesAmount + 1;

}while(commission<goal);

System.out.println(salesAmount);

}

}

Output:

Graphical user interface, application, Word

Description automatically generated

1. (No Mistakes)
2. (No Mistakes)