COMP101 LAB8

# Requirements

Make a Java program that calculates and displays information about penalties for late submission for a piece of coursework as described below.

The piece of coursework is given a mark that is an integer between 0 and 100.

The mark for the piece of coursework to pass is 40%. Each piece of coursework may be subject

to a late penalty if it is submitted after the deadline. There are two ways of calculating late

penalties.

Scheme 1

The work loses 5 marks every day it is late. If the mark drops below 20, no further

late deductions are made and the mark stays as 20. If the original mark is initially less

than or equal to 20, then the mark stays as its original value.

Scheme 2

The mark for the work is reduced by 10% of the current mark every day it is late.

If the mark drops below 25, no further late deductions are made and the mark stays

as 25. If the original mark is initially less than or equal to 25, the mark stays as its

original value. http://www.gearbest.com/led-strips/pp\_161659.html

It should also output the number of days until the work no longer passes

# Analysis and Design

There will be 2 classes within the solution, a markCalculator class and a MarkUser class, there is also a MarkUserExt class for the extended requirements, the MarkCalculator Class will contain all the calculations and method

## PseudoCode

### Mark Calculator class

INT mark

INT days

INT passMark = 40

FUNCTION setDays

INPUT INT (X)

OUTPUT VOID

days = X

END

FUNCTION setMark

INPUT INT (X)

OUTPUT VOID

mark = X

END

FUNCTION setPassMark

INPUT INT (X)

OUTPUT VOID

passMark = X

END

FUNCTION inputBetween

INPUT INT (min) INT (max)

OUTPUT INT

INT value = min-1

WHILE value < min OR value > max

value = INT INPUT

IF value value < min OR value > max

OUTPUT The mark must be between min and max, enter a new value:

ELSE

BREAK

END LOOP

RETURN value

CLASS schemeOne

INT lowerLimit = 20

INT marksLostPerDay = 5

FUNCTION setLowerLimit

INPUT INT (X)

OUTPUT VOID

lowerLimit = X

END

FUNCTION setMarksLostPerDay

INPUT INT (X)

OUTPUT VOID

marksLostPerDay = X

END

FUNCTION getMarks

INPUT VOID

OUTPUT INT ARRAY

INT currentMark = mark

INT ARRAY markList

FOR day = 0; day <= days; day++

markList[day] = currentMark;

currentMark -= marksLostPerDay

IF currentMark < lowerLimit AND mark>lowerLimit

currentMark = lowerLimit

ELSE IF mark < lowerLimit

currentMark = mark

RETURN markList

FUNCTION getDays

INPUT VOID

OUTPUT INT

RETURN (mark-passMark)/marksLostPerDay

FUNCTION printMarks

INPUT INT ARRAY (markList)

OUTPUT VOID

OUTPUT "day | mark"

FOR day = 0; day <= markList.length-1; day++

OUTPUT day | markList[day]

CLASS schemeOne

INT lowerLimit = 25

INT percentageLostPerDay = 10

FUNCTION setLowerLimit

INPUT INT (X)

OUTPUT VOID

lowerLimit = X

END

FUNCTION setPercentageLostPerDay

INPUT INT (X)

OUTPUT VOID

percentageLostPerDay = X

END

FUNCTION getMarks

INPUT VOID

OUTPUT DOUBLE ARRAY

DOUBLE currentMark = mark

DOUBLE ARRAY markList

FOR day = 0; day <= days; day++

markList[day] = currentMark;

currentMark = currentMark\*(100-percentageLostPerDay)/100

IF currentMark < lowerLimit AND mark>lowerLimit

currentMark = lowerLimit

ELSE IF mark < lowerLimit

currentMark = mark

RETURN markList

FUNCTION getDays

INPUT VOID

OUTPUT INT

DOUBLE marksRemaining = mark

INT day = -1

WHILE marksRemaining > passMark

marksRemaining = marksRemaining\*(100-percentageLostPerDay)/10

day ++

END loop

RETURN day

FUNCTION printMarks

INPUT DOUBLE ARRAY (markList)

OUTPUT VOID

OUTPUT "day | mark"

FOR day = 0; day <= markList.length-1; day++

OUTPUT day | markList[day]

END

## Class Diagram

|  |  |  |
| --- | --- | --- |
| TunnelTollCostUser |  | Vehicle |
|  | int: wheels  double: length  int: axles  double: weight  int: vehicleClass  scanner: input |
| main() |
| |  | | --- | | TunnelTollCost | | boolean: isWeekend | | double: getCharge(int vehicleClass)  double: getMonthlyPass(int vehicleClass)  double: getMonthlyCost(int vehicleClass, int weekendJourneys, int weekdayJourneys) | | int :getIntInput(int minValue)  double: getDoubleinput(double minValue)  void: inputWheels()  void: inputLength()  void: inputAxles()  void: inputWeight()  int: identify()  void: setClass(int vehicleClass)  void: identifyClass() |

# Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Wheels | Length | Axles | Weight | Weekend | Vehicle Class | Cost | As Expected |
| 2 |  |  |  | True | 1 | 1.50 | Yes |
| 4 | 7 |  |  | True | 2 | 2.40 | Yes |
| 6 | 16 | 3 | 2 | True | 3 | 2.85 | Yes |
| 4 | 17 | 2 |  | False | 4 | 4.40 | Yes |
| 4 | 22 | 4 | 2.2 | False | 5 | 8.00 | Yes |
| 8 | 33 | 4 | 3.6 | False | 6 | 12.00 | Yes |
| 1 | Asks for greater number to be input | | | | | | Yes |

Please enter the number of wheels: 8

Please enter the Vehicle Length: 33

Please enter the Number of axles: 4

Please enter the Vehicle Weight: 3.6

Input true if travelling on a weekend (sat/sun) and false otherwise: false

A Large Lorry (Class 6) traveling on a weekday will be charged 12.00

Please enter the number of wheels: 1

The number must be greater than or equal to to 2

Please enter a different number: