Elias Zalles

ITP 100

Module 2 Chapter 4

Q1.

// Global constants for hourly calculation

Constant Integer HOURS = 40

Constant Real OT\_MULTIPLIER = 1.5

// Main Module

Module Main ()

// Local variables

Declare weekly pay

// Get the numbers of hours worked

Call gethoursWork (hoursWork)

// Get the job code

Call getjobCode (jobCode)

// Call the calcWeekly module with the week’s pay.

Call calcWeekly

End Module

// getHoursWork receives an input for the number of hours worked and stores it into

// hoursWork parameter

Module getHoursWork (Real Ref hoursWork)

Declare Real hoursWork

Display “Enter the number of hours worked.”

Input hours

End Module

// getJobCode receives an input for the job code and stores it into the jobCode parameter

getJobCode (String Ref jobCode)

Declare String jobCode

Display “Enter your job code.”

Input jobCode

End Module

// Module calcPay calculates overtime hours. Standard hours are stored in regHours, overtime is

// stored in otHours.

calcPay (Real Ref hoursWork)

Declare Real otHours, regHours

If hoursWork > BASE\_HOURS Then

Set otHours = BASE\_HOURS – 40

Else

Set regHours = hoursWork

End If

End Module

// calcWeekly will pull the hours worked, including OT, and the job code to

// determine weekly pay.

calcWeekly(Real Ref otHours, Real Ref regHours)

Declare Real Rate\_P, Rate\_Q, Rate\_R, Rate\_S

Select jobCode

Case P:

Set Rate\_P = (regHours \* 10) + otHours(10 \* 1.5)

Case Q:

Set Rate\_Q = (regHours \* 12.50) + otHours(12.50 \* 1.5)

Case R:

Set Rate\_R = (regHours \* 15) + otHours(15 \* 1.5)

Case S:

Set Rate\_S = (regHours \* 20) + otHours(20 \* 1.5)

Default:

Display “Invalid Job Code, please try again.”

End Select

End Module

Q2.

// Global constants

Constant Real CREDIT\_HOUR = 500

// Main Module

Module Main ()

// Local variables

Declare Real cost

Declare Real discount

// Get the discount percentage

Call getdiscountPercent (discountPercent)

// Get the tuition cost

Call gettuitionCost (tuitionCost)

End Module

// getdiscountPercent receives an input for the discount percentage and stores it into

// discountPercent parameter

Module get discountPercent (Real Ref discountPercent)

Declare Real discountPercent

Display “Enter the discount percentage.”

Input discount

End Module

// gettuitionCost receives an input for the tuition cost and stores it into the tuitionCost parameter

gettuitionCost (String Ref tuitionCost)

Declare String tuitionCost

Display “Enter your tuiton cost.”

Input tuitionCost

End Module

// Module discountCost calculates cost after the discount. Standard cost are stored in tuitionCost, is

// stored in discountPercent.

discountCost (Real Ref tuitionCost)

Declare Real discountCost

If tuitionCost > CREDIT\_HOUR Then

Set discountCost= CREDIT\_HOUR – 500

Else

Set tuitionCost = discountCost

End If

End Module

// discountPercent will pull the discount, including tuition, and the tuition cost to

// determine cost after discount

discountCost Real Ref tuitonCost, Real Ref discountPercent)

Declare Real 16, 15, 10, 5

Select discountCost

Else If credits <= above AND credits => 16

Then discount = .1

Else If credits <= 15 AND credits => 11

Then discount = .08

Else If credits <= 10 AND credits => 6

Then discount = .05

Else If credits <= 5 AND credits => 1

Then discount = .02

Else Display "Invalid number of credits entered."

End If

Set tuition = credits \* fee

Display "Total tuition = $" & tuition

Set discountPercent = discount \* 100

Display "Discount percentage = " & dicountPercent & "%"

Set tuitionCost = tuition - (tuition \* discount)

Display "Tuition cost= $" & tuitionCost

End Module

Q3.

// Global constants

// Main Module

Module Main ()

// Declare Local variables

Declare Real mealprice

Declare Real tip

Declare Real tax

Declare Real total

//Function calls

Call input\_meal(mealprice)

Call calc\_tip(mealprice, tip)

Call calc\_tax(mealprice, tax)

Call calc\_total(mealprice, total, tip, tax)

Call print\_info(mealprice, total, tip, tax)

End Module

//this function will input meal price

Module input\_meal(Real Ref mealprice)

Display “Enter the meal price $”

Input mealprice

End Module

//this function will calculate tip at different prices

Module calc\_tip(Real mealprice, Real Ref tip)

If mealprice >= .01 AND mealprice <=5.99 Then

Set tip = mealprice \* .10

Else If mealprice >=6 AND mealprice <=12 Then

Set tip = mealprice \* .13

Else If mealprice >=12.01 AND mealprice <=17 Then

Set tip = mealprice \* .16

Else If mealprice >= 17.01 and mealprice <= 25 Then

Set tip = mealprice \* .19

Else

tip = mealprice \* .22

End If

End Module

// Module calc\_tax will calculate tax at 6%

Module calc\_tax(Real mealprice, Real Ref tax)

Set tax = mealprice \* .06

End Module

// Module calc\_total will calculate tip, tax, and the total cost

Module calc\_total(Real mealprice, Real Ref total, Real tip, Real tax)

Set total = mealprice + tip + tax

End Module

// print\_info will print tip, tax, the mealprice, and the total

Module print\_info(Real mealprice, Real total, Real tip, Real tax)

Display “The meal price is $”, mealprice

Display “The tip is $”, tip

Display “The tax is $”, tax

Display “The total is $”, total

End Module