**Community Garden Appointment Program Pseudocode**

**By: Alis Marquez and Saul Sanchez**

**START** Main.java

IMPORT java.util.\*

CLASS Main

METHOD main(String args[])

CREATE SCANNER intInput

CREATE SCANNER name

CREATE SCANNER phone

CREATE SCANNER email

CREATE BOOLEAN validT

PRINT “Welcome to the Community Garden Appointment Service.”

PRINT “Full Name: “

INPUT n = name

PRINT “Phone Number: “

INPUT p = phone

PRINT “E-Mail: “

INPUT e = email

CREATE BOOKTIME appt

CALL printBlock()

INPUT inputMonth = intInput

CALL validate(inputMonth)

CALL printBlockDay(inputMonth)

INPUT inputDay = intInput

CALL validateDay(inputDay)

DO

CALL printBlockTime(inputDay)

INPUT inputTime = intInput

SET validT = CALL validateTime(inputDay, inputTime)

WHILE validT is false

CREATE DONATIONS d

CREATE DOUBLE total = 0.0

CALL printBlockDono()

INPUT inputDono = intInput

SWITCH inputDono

CASE 1

PRINT “Each pound of seeds is $30.00. How many pounds would you like to donate?”

INPUT numSeeds = intInput

CALL setAmount(numSeeds)

SET total = CALL calcSeeds()

BREAK

CASE 2

PRINT “Each pound of soil is $15.00. How many pounds would you like to donate?”

INPUT numSoil = intInput

CALL setAmount(numSoil)

SET total = CALL calcSoil()

BREAK

CASE 3

PRINT “Maybe next time!”

BREAK

DEFAULT

PRINT “Invalid input. Please try again.”

ENDSWITCH

PRINT “User Details”

CALL printDetails()

PRINT “You chose: “

CALL print()

PRINT “Total Due: $” + total

ENDMETHOD

ENDCLASS

**END** Main.java

**START** BookMonth.java

CLASS BookMonth implements Display and Validation interfaces

CREATE STRING month

CREATE INT day

CONSTRUCTOR BookMonth()

SET month = “ “

ENDCONSTRUCTOR

OVERLOADED CONSTRUCTOR BookMonth(String)

month equals String argument

ENDCONSTRUCTOR

METHOD setMonth(String)

SET month = String argument

ENDMETHOD

METHOD getMonth()

RETURN String month

ENDMETHOD

METHOD setDay(int)

SET day = int argument

ENDMETHOD

METHOD getDay()

RETURN int day

ENDMETHOD

METHOD print()

PRINT “Month: “ + month

PRINT “Day: “ + day

ENDMETHOD

METHOD printBlock()

CREATE STRING months[] = 12 month names

PRINT “Please pick a month for your appointment.”

FOR all 12 months

PRINT one months[] element

ENDFOR

PRINT “Enter a number: “

ENDMETHOD

METHOD printBlockDay(int)

CREATE INT months[12][]

SET months[element] = number of days per month

PRINT “Please pick a day for your appointment.”

PRINT “S M T W T F S”

CREATE INT dayNum = 1

FOR the month chosen

IF day is one digit, add zero before it

PRINT “0”

ENDIF

PRINT dayNum + “ “

INCREMENT dayNum

IF loop reaches the day after the 7th day

PRINT new line

ENDIF

ENDFOR

ENDMETHOD

METHOD validate(int)

CREATE STRING m

CREATE STRING months[] = 12 month names

IF input is 1 or higher AND 12 or lower

SET m = months[input - 1]

CALL setMonth(m)

ENDIF

ELSE

PRINT “Input error. Try again.”

EXIT

ENDELSE

ENDMETHOD

METHOD validateDay(int)

IF input is 1 or higher AND 31 or lower

CALL setDay(input)

ENDIF

ELSE

PRINT “Input error. Try again.”

EXIT

ENDELSE

ENDMETHOD

ENDCLASS

**END** BookMonth.java

**START** BookTime.java

IMPORT java.util.\*

CLASS BookTime extends BookMonth

CREATE STRING time

CONSTRUCTOR BookTime()

CALL super()

SET time = “ “

ENDCONSTRUCTOR

METHOD setTime(String)

SET time = String argument

ENDMETHOD

METHOD getTime()

RETURN time

ENDMETHOD

METHOD print()

CALL super.print()

PRINT “Time: “ + time

ENDMETHOD

METHOD printBlockTime(int)

CREATE STRING times[] = 1 hour intervals from 8am to 8pm

IF input is a Sunday

CREATE SCANNER i

PRINT “The garden is not open on Sundays. Please make another selection.”

INPUT newInput = i

CALL printBlockTime(newInput)

ENDIF

ELSE IF input is a Saturday

PRINT “You chose a Saturday. Please pick from the times available.”

FOR every hour

PRINT one times[] element

IF i variable reaches element 9

SET i = length of array

ENDIF

ENDFOR

ENDELSEIF

ELSE

PRINT “You chose a weekday. Please pick from the times available.”

FOR ever hour

PRINT one times[] element

ENDFOR

ENDELSE

ENDMETHOD

METHOD valdateTime(int, int)

CREATE STRING times[] = 1 hour intervals from 8am to 8pm

IF day input is a Saturday

IF time input is less than 1 OR greater than 10

RETURN false

ENDIF

ENDIF

ELSE

IF time input is less than 1 OR greater than 13

RETURN false

ENDIF

ENDELSE

PRINT “ “

CALL setTime(times[time input - 1])

RETURN true

ENDMETHOD

ENDCLASS

**END** BookTime.java

**START** Donations.java

CLASS Donations

CREATE INT amount

CREATE DOUBLE total

CONSTRUCTOR Donations()

SET amount = 0

SET total = 0.0

ENDCONSTRUCTOR

METHOD setAmount(int)

SET amount = int argument

ENDMETHOD

METHOD getAmount()

RETURN amount

ENDMETHOD

METHOD setTotal(double)

SET total = double argument

ENDMETHOD

METHOD getTotal()

RETURN total

ENDMETHOD

METHOD calcSeeds()

RETURN total = amount \* 30

ENDMETHOD

METHOD calcSoil()

RETURN total = amount \* 15

ENDMETHOD

METHOD printBlockDono()

PRINT “Before you leave, would you like to make a donation?”

PRINT “Please choose an option.”

PRINT “1. Seeds”

PRINT “2. Soil”

PRINT “3. Not right now”

ENDMETHOD

ENDCLASS

**END** Donations.java

**START** UserDetails.java

CLASS UserDetails

CREATE STRING name

CREATE STRING phone

CREATE STRING email

CONSTRUCTOR UserDetails()

SET name = “ “

SET phone = “ “

SET email = “ “

ENDCONSTRUCTOR

OVERLOADED CONSTRUCTOR UserDetails(String, String, String)

SET name = 1st String argument

SET phone = 2nd String argument

SET email = 3rd String argument

ENDCONSTRUCTOR

METHOD setDetails(String, String, String)

SET name = 1st String argument

SET phone = 2nd String argument

SET email = 3rd String argument

ENDMETHOD

METHOD printDetails()

PRINT “Name: “ + name

PRINT “Phone Number: “ + phone

PRINT “E-Mail: “ + email

PRINT new line

ENDMETHOD

ENDCLASS

**END** UserDetails.java

**START** Display.java

INTERFACE Display

METHOD print()

METHOD printBlock()

ENDINTERFACE

**END** Display.java

**START** Validation.java

INTERFACE Validation

METHOD validate(int)

ENDINTERFACE

**END** Validation.java