Pseudo Code

Main

Declare class objects that will be needed throughout program

Call the setInfo() function to begin user registration

Begin a do-while loop and do the following

Call the options() function to give the user a list of options and determine what they want to do

Check user input with each case within the switch statement

If case 1, call the setAppt() function to begin scheduling an appointment

If case 2, begin another do-while loop and switch statement, and do the following

If case 1, call mon.setDonation to begin a monetary donation

If case 2, call mat.setDonation to being a material donation.

If case 3, return to the options menu

If user input matches none of the cases, call dError() to display a donation error and begin loop again

Continue until one the cases is met

If case 3, call mat.cancel() function to cancel a material donation.

If case 4, call a.cancel() to cancel appointment

If case 5, exit the program

If user input does not meet any case, tell them that and repeat loop.

End of Main

Appointment

Declare all required class members

Define class constructor which will be used to initialize several members

Begin method signature of setDay()

Declare local variables to be used in this method

Begin for-loop to determine limit of dates for each month of the year

Check the current index to determine what month is currently being filled

This will determine the cap on how high the dates can reach, i.e. 31 for Jan. and 28 for Feb.

Begin inner for-loop to start randomly generating dates for the current month

Begin a do-while and check to see if the date generated has already been added by calling chckDay()

If it has, repeat the previous step

If not, add it to the month

Once the month has been filled, sort it from least to greatest by calling sort()

End of setDay() signature

Begin method signature of sort()

Begin a for-loop to go through the current month and sort the dates

Beginning with the first date (index 0), if there is a date that is less than it, swap the values in those indices.

Call the Sunday() method to determine what day the date falls on and then match it with that date

End of sort() signature

Begin method signature of chckDay()

Begin a for-loop to go through all of the dates already added to the current month and check to see if the current randomly generated date has already been added.

If it has, return to setDay() and obtain a new date

If the control has not been returned to setDay(), call Sunday() to see if the date would fall on a Sunday.

If it does, return to setDay() to obtain a new date

Otherwise, add the date to the current month and return to setDay() to get begin loop again

End of chckDay()

Begin method signature of Sunday()

Begin a while loop that will iterate until a local variable declared as count is equal to the current date, i.e. the 17th = 0-16

Add 1 to a local variable with each iteration

If the variable equals 6, set it to 0

Increment count

Repeat loop

Once loop is done, if the local variable is equal to 6, return to setDate() and get a new date

End of Sunday()

Begin method signature of setAppt()

Begin a Do-While loop

Ask user to enter the month in which they'd like to visit

Repeat until user enters a valid month

End of setAppt()

Begin method signature of chckMth()

Check the users input for month selection and compare it to all 12 months

When a match is found, call getDate() to display all available dates

If no match is found, tell the user that and return to previous function call

End of chckMth()

Begin method signature of getDate()

Begin a Do-While loop

Display all available dates for the chosen month

Ask the user for input and compare to available dates

If match is found, schedule the appointment

Else, repeat until input matches available date

End of getDate()

Begin method signature of getTime()

Show the user the available hours for the day

If the day is Mon-Fri, times will be from 8am to 7pm

If day is Saturday, times will be from 8am to 4pm

Ask user to input the time they want to come and visit

Repeat until valid time is entered

End of getTime()

Begin method signature of cancel()

If the class member meant to hold the finalized appointment date is empty, no appointment has been scheduled and thus cannot be canceled

Tell user this

Else, verify that the user wants to cancel

If they do, cancel the appointment and make the file that contained the appointment details become empty

Else cancel the cancellation

End of getTime()

Begin method signature of toFile()

Determine whether an appointment has been scheduled

If it is, make the file empty

Else, write the details of the appointment to the file

End of toFile()

End of Appointment

Material

Declare class members to be used

Initialize several methods in the constructor

Begin method signature of getType()

Return the type of donation. In this case, material.

End of getType()

Begin method signature of setDonation()

Show the user a list of items the Garden is currently accepting

Get user input and compare it to the list

If input matches, set the donation

If user wants to cancel, cancel the donation

Repeat until either is met

End of setDonation()

Begin method signature of getDonation()

If no donation has been made, tell the user this

Else return the donation item

End of getDonation()

Begin method signature of cancel()

If no donation has been made, tell the user this

Else cancel the donation

End of cancel()

Begin method signature of toFile()

Determine whether an appointment has been scheduled

If it is, make the file empty

Else, write the details of the appointment to the file

End of toFile()

Begin method signature of blank()

Return false if a donation has been made

Else return true

End of signature()

End of Material

Monetary

Declare class members

Initialize some in the constructor

Begin method signature of setDonation()

Begin a Do-While loop

Ask the user how much they want to donate

Tell them only exact amounts are accepted like $27

Check if input is an exact amount

If it is, set the donation

Else tell them input was invalid and repeat loop

End of setDonation()

Begin method signature of getType()

Return donation type. (Material)

End of getType()

Begin method signature of getDonation()

Check if donation has been made

If one has, return the amount

Else tell the user no donation has been made

End of getDonation()

Begin method signature of blank()

Return true if no donation has been made

Else return false

End of blank()

Begin method signature of toFile()

Determine whether an appointment has been scheduled

If it is, make the file empty

Else, write the details of the appointment to the file

End of toFile()

End of Material

Contact Info

Declare class members

Begin method signature of setInfo()

Prompt the user to enter general registration info

End of setInfo()

Begin method signature of getName()

Return name on record

End getName()

Being method signature of getPhone()

Return phone number on record

End of getPhone()

Begin method signature of getEmail()

Return email on record

End of getEmail()

End of Contact Info

Menu

Begin method signature of Options()

Prompt the user with a list of options

Get user input and return it to main

End of Options()

Begin method signature of donation()

Prompt the user to decide what kind of donation they want to make

Return input to main

End of donation()

Begin method signature of dError()

Display invalid input has been entered to user if they do not enter correct donation type

End of dError()

End of Menu