Desktop Application Programming

Joheen Chakraborty

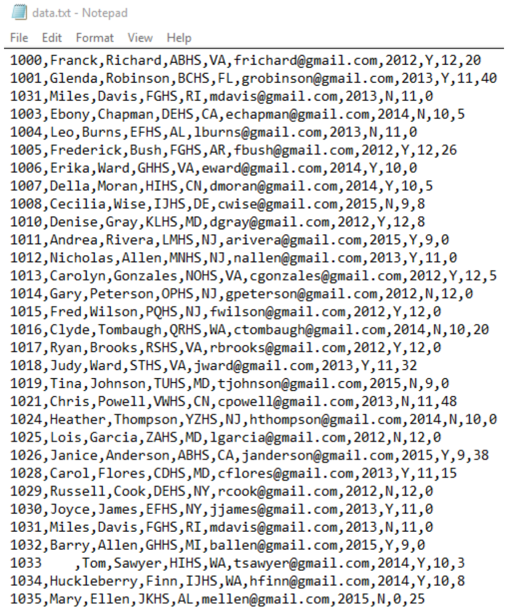
John P. Stevens High School, Edison, NJ

The Problem

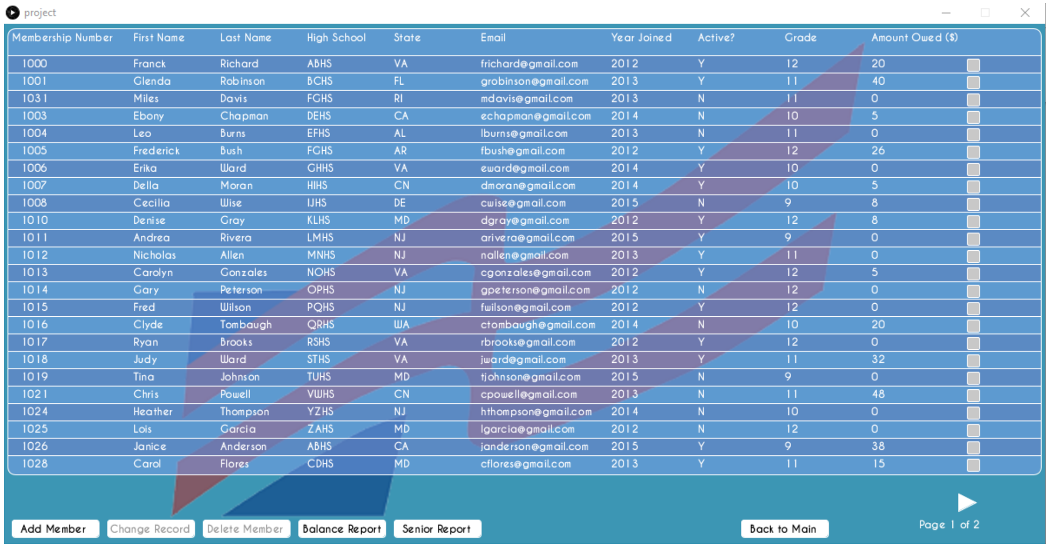
- Hired by national office of FBLA-PBL
- Needs a program to manage membership data
  – add, delete, change members
- Needs two reports
  1. Balance Report
  2. Senior Report
- Should be easy to navigate and user-friendly

Why is the program useful?

- Organizes large amounts of data
- Allows easy manipulation
- Easy to sort members and create reports
- More user-friendly than a text file or database



```
1000,Franck,Richard,ABHS,VA,frichard@gmail.com,2012,Y,12,20
1001,Glenda,Robinson,BCHS,FL,grobinson@gmail.com,2013,Y,11,40
1031,Miles,Davis,FGHS,RI,mdavis@gmail.com,2013,N,11,0
1003,Ebony,Chapman,DEHS,CA,echapman@gmail.com,2014,N,10,5
1004,Leo,Burns,EFHS,AL,lburns@gmail.com,2013,N,11,0
1005,Frederick,Bush,FGHS,AR,fbush@gmail.com,2012,Y,12,26
1006,Erika,Ward,GHHS,VA,eward@gmail.com,2014,Y,10,0
1007,Della,Moran,HIHS,CN,dmoran@gmail.com,2014,Y,10,5
1008,Cecilia,Wise,IJHS,DE,cwise@gmail.com,2015,N,9,8
1010,Denise,Gray,KLHS,MD,dgray@gmail.com,2012,Y,12,8
1011,Andrea,Rivera,LMHS,NJ,arivera@gmail.com,2015,Y,9,0
1012,Nicholas,Allen,MNHS,NJ,nallen@gmail.com,2013,Y,11,0
1013,Carolyn,Gonzales,NOHS,VA,cgonzales@gmail.com,2012,Y,12,5
1014,Gary,Peterson,OPHS,NJ,gpeterson@gmail.com,2012,N,12,0
1015,Fred,Wilson,PQHS,NJ,fwilson@gmail.com,2012,Y,12,0
1016,Clyde,Tombaugh,QRHS,WA,ctombaugh@gmail.com,2014,N,10,20
1017,Ryan,Brooks,RSHS,VA,rbrooks@gmail.com,2012,Y,12,0
1018,Judy,Ward,STHS,VA,jward@gmail.com,2013,Y,11,32
1019,Tina,Johnson,TUHS,MD,tjohnson@gmail.com,2015,N,9,0
1021,Chris,Powell,VWHS,CN,cpowell@gmail.com,2013,N,11,48
1024,Heather,Thompson,YZHS,NJ,hthompson@gmail.com,2014,N,10,0
1025,Lois,Garcia,ZAHS,MD,lgarcia@gmail.com,2012,N,12,0
1026,Janice,Anderson,ABHS,CA,janderson@gmail.com,2015,Y,9,38
1028,Carol,Flores,CDHS,MD,cflores@gmail.com,2013,Y,11,15
1029,Russell,Cook,DEHS,NY,rcook@gmail.com,2012,N,12,0
1030,Joyce,James,EFHS,NY,jjames@gmail.com,2013,Y,11,0
1031,Miles,Davis,FGHS,RI,mdavis@gmail.com,2013,N,11,0
1032,Barry,Allen,GHHS,MI,ballen@gmail.com,2015,Y,9,0
1033    ,Tom,Sawyer,HIHS,WA,tsawyer@gmail.com,2014,Y,10,3
1034,Huckleberry,Finn,IJHS,WA,hfinn@gmail.com,2014,Y,10,8
1035,Mary,Ellen,JKHS,AL,mellen@gmail.com,2015,N,0,25
```



Membership Number	First Name	Last Name	High School	State	Email	Year Joined	Active?	Grade	Amount Owed ($)	
1000	Franck	Richard	ABHS	VA	frichard@gmail.com	2012	Y	12	20	☐
1001	Glenda	Robinson	BCHS	FL	grobinson@gmail.com	2013	Y	11	40	☐
1031	Miles	Davis	FGHS	RI	mdavis@gmail.com	2013	N	11	0	☐
1003	Ebony	Chapman	DEHS	CA	echapman@gmail.com	2014	N	10	5	☐
1004	Leo	Burns	EFHS	AL	lburns@gmail.com	2013	N	11	0	☐
1005	Frederick	Bush	FGHS	AR	fbush@gmail.com	2012	Y	12	26	☐
1006	Erika	Ward	GHHS	VA	eward@gmail.com	2014	Y	10	0	☐
1007	Della	Moran	HIHS	CN	dmoran@gmail.com	2014	Y	10	5	☐
1008	Cecilia	Wise	IJHS	DE	cwise@gmail.com	2015	N	9	8	☐
1010	Denise	Gray	KLHS	MD	dgray@gmail.com	2012	Y	12	8	☐
1011	Andrea	Rivera	LMHS	NJ	arivera@gmail.com	2015	Y	9	0	☐
1012	Nicholas	Allen	MNHS	NJ	nallen@gmail.com	2013	Y	11	0	☐
1013	Carolyn	Gonzales	NOHS	VA	cgonzales@gmail.com	2012	Y	12	5	☐
1014	Gary	Peterson	OPHS	NJ	gpeterson@gmail.com	2012	N	12	0	☐
1015	Fred	Wilson	PQHS	NJ	fwilson@gmail.com	2012	Y	12	0	☐
1016	Clyde	Tombaugh	QRHS	WA	ctombaugh@gmail.com	2014	N	10	20	☐
1017	Ryan	Brooks	RSHS	VA	rbrooks@gmail.com	2012	Y	12	0	☐
1018	Judy	Ward	STHS	VA	jward@gmail.com	2013	Y	11	32	☐
1019	Tina	Johnson	TUHS	MD	tjohnson@gmail.com	2015	N	9	0	☐
1021	Chris	Powell	VWHS	CN	cpowell@gmail.com	2013	N	11	48	☐
1024	Heather	Thompson	YZHS	NJ	hthompson@gmail.com	2014	N	10	0	☐
1025	Lois	Garcia	ZAHS	MD	lgarcia@gmail.com	2012	N	12	0	☐
1026	Janice	Anderson	ABHS	CA	janderson@gmail.com	2015	Y	9	38	☐
1028	Carol	Flores	CDHS	MD	cflores@gmail.com	2013	Y	11	15	☐

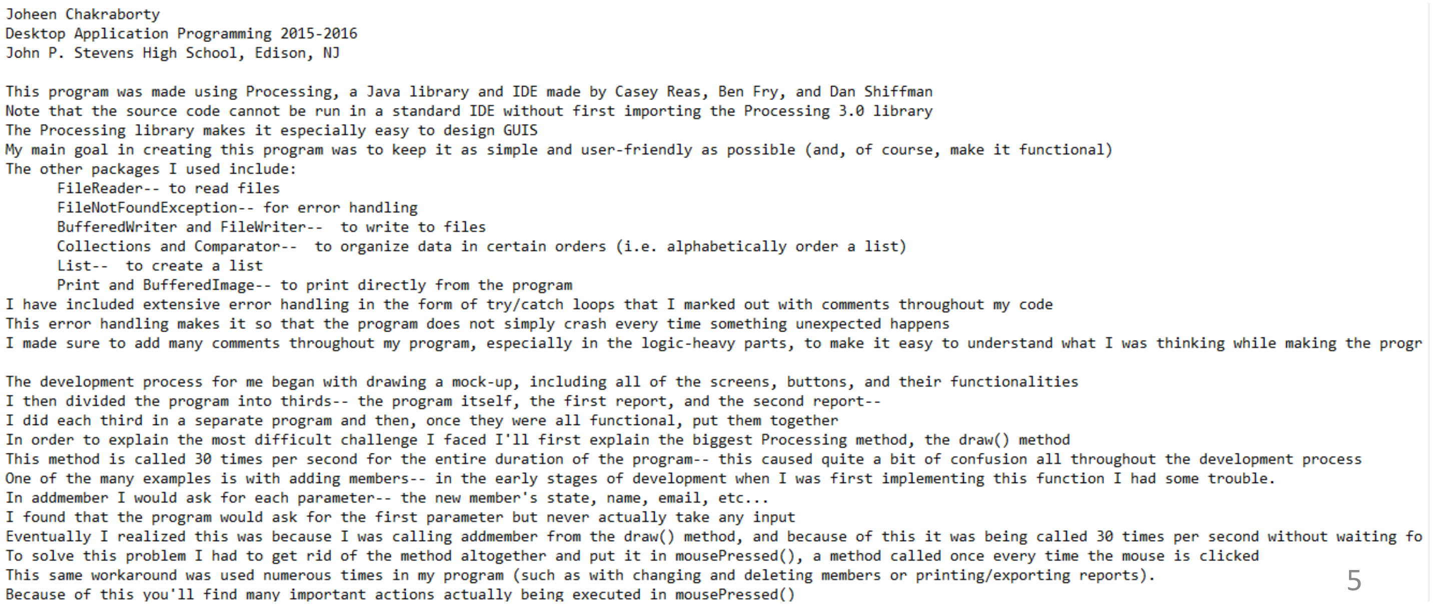
Program Documentation: Open Source Components

- Processing 3.0.1 (with Java)

  https://processing.org/
- Java libraries (https://docs.oracle.com/javase/7/docs/api/):
  - Filereader
  - Bufferedwriter, Filewriter
  - Collections, Comparator
  - Print, BufferedImage

Program Documentation: Components Developed

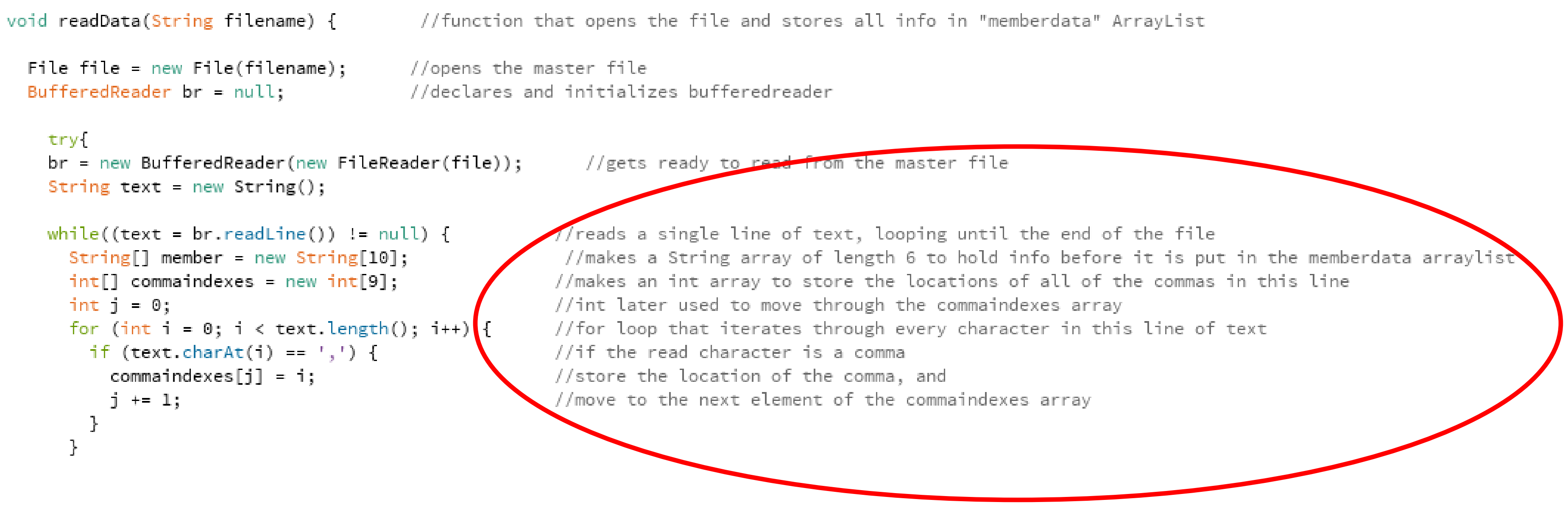
- readme.txt
  - purpose of the program
  - software used for development
  - instructions for running the program



```
Joheen Chakraborty
Desktop Application Programming 2015-2016
John P. Stevens High School, Edison, NJ

This program was made using Processing, a Java library and IDE made by Casey Reas, Ben Fry, and Dan Shiffman
Note that the source code cannot be run in a standard IDE without first importing the Processing 3.0 library
The Processing library makes it especially easy to design GUIS
My main goal in creating this program was to keep it as simple and user-friendly as possible (and, of course, make it functional)
The other packages I used include:
      FileReader-- to read files
      FileNotFoundException-- for error handling
      BufferedWriter and FileWriter--  to write to files
      Collections and Comparator--  to organize data in certain orders (i.e. alphabetically order a list)
      List--  to create a list
      Print and BufferedImage-- to print directly from the program
I have included extensive error handling in the form of try/catch loops that I marked out with comments throughout my code
This error handling makes it so that the program does not simply crash every time something unexpected happens
I made sure to add many comments throughout my program, especially in the logic-heavy parts, to make it easy to understand what I was thinking while making the progr

The development process for me began with drawing a mock-up, including all of the screens, buttons, and their functionalities
I then divided the program into thirds-- the program itself, the first report, and the second report--
I did each third in a separate program and then, once they were all functional, put them together
In order to explain the most difficult challenge I faced I'll first explain the biggest Processing method, the draw() method
This method is called 30 times per second for the entire duration of the program-- this caused quite a bit of confusion all throughout the development process
One of the many examples is with adding members-- in the early stages of development when I was first implementing this function I had some trouble.
In addmember I would ask for each parameter-- the new member's state, name, email, etc...
I found that the program would ask for the first parameter but never actually take any input
Eventually I realized this was because I was calling addmember from the draw() method, and because of this it was being called 30 times per second without waiting fo
To solve this problem I had to get rid of the method altogether and put it in mousePressed(), a method called once every time the mouse is clicked
This same workaround was used numerous times in my program (such as with changing and deleting members or printing/exporting reports).
Because of this you'll find many important actions actually being executed in mousePressed()
```

Program Documentation: Components Developed

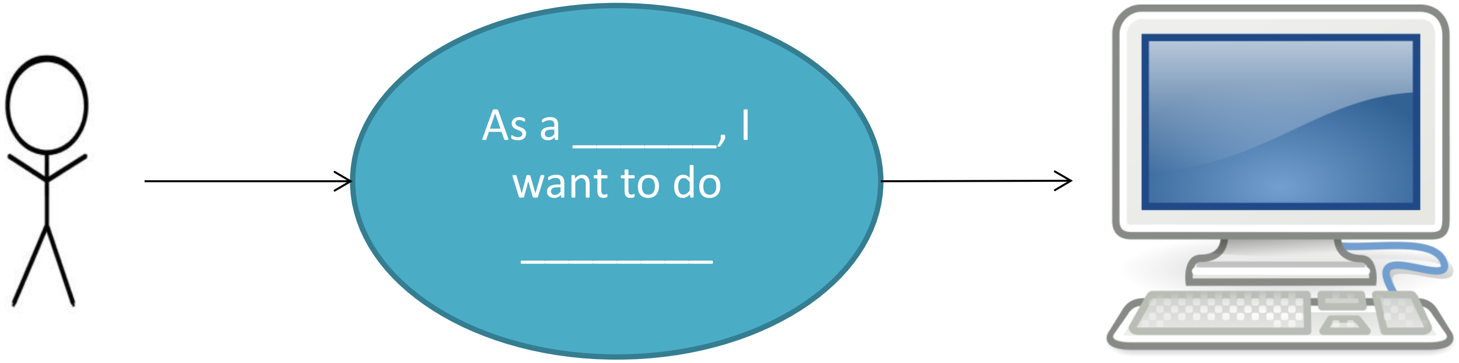
- Source code
  - – comments explaining program logic

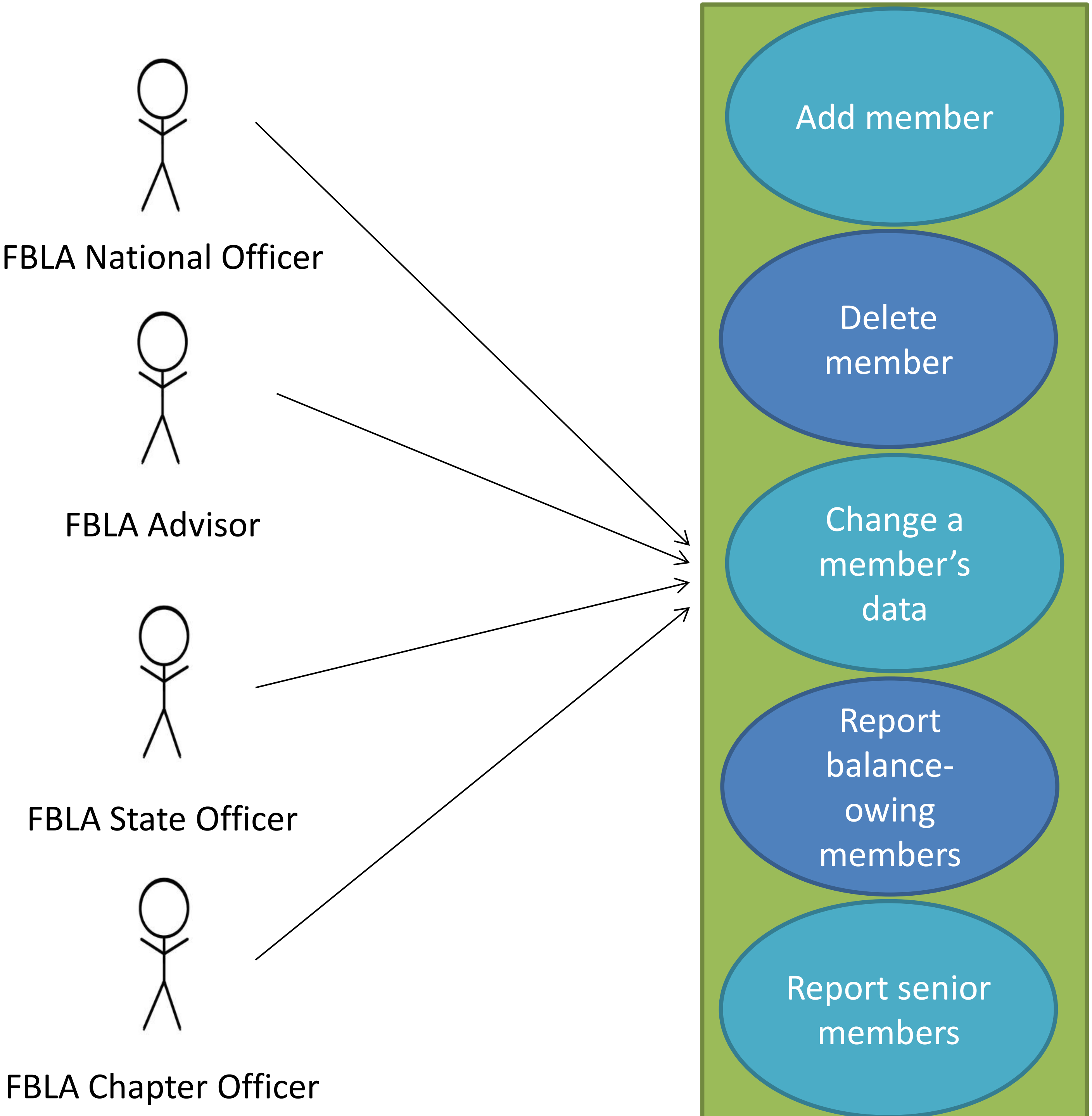


```
void readData(String filename) {        //function that opens the file and stores all info in "memberdata" ArrayList

  File file = new File(filename);      //opens the master file
  BufferedReader br = null;            //declares and initializes bufferedreader

    try{
    br = new BufferedReader(new FileReader(file));      //gets ready to read from the master file
    String text = new String();

    while((text = br.readLine()) != null) {          //reads a single line of text, looping until the end of the file
      String[] member = new String[10];               //makes a String array of length 6 to hold info before it is put in the memberdata arraylist
      int[] commaindexes = new int[9];               //makes an int array to store the locations of all of the commas in this line
      int j = 0;                                     //int later used to move through the commaindexes array
      for (int i = 0; i < text.length(); i++) {      //for loop that iterates through every character in this line of text
        if (text.charAt(i) == ',') {                 //if the read character is a comma
          commaindexes[j] = i;                       //store the location of the comma, and
          j += 1;                                    //move to the next element of the commaindexes array
        }
      }
```

Planning Process

- Actors
- Use cases
- User stories





FBLA National Officer

FBLA Advisor

FBLA State Officer

FBLA Chapter Officer

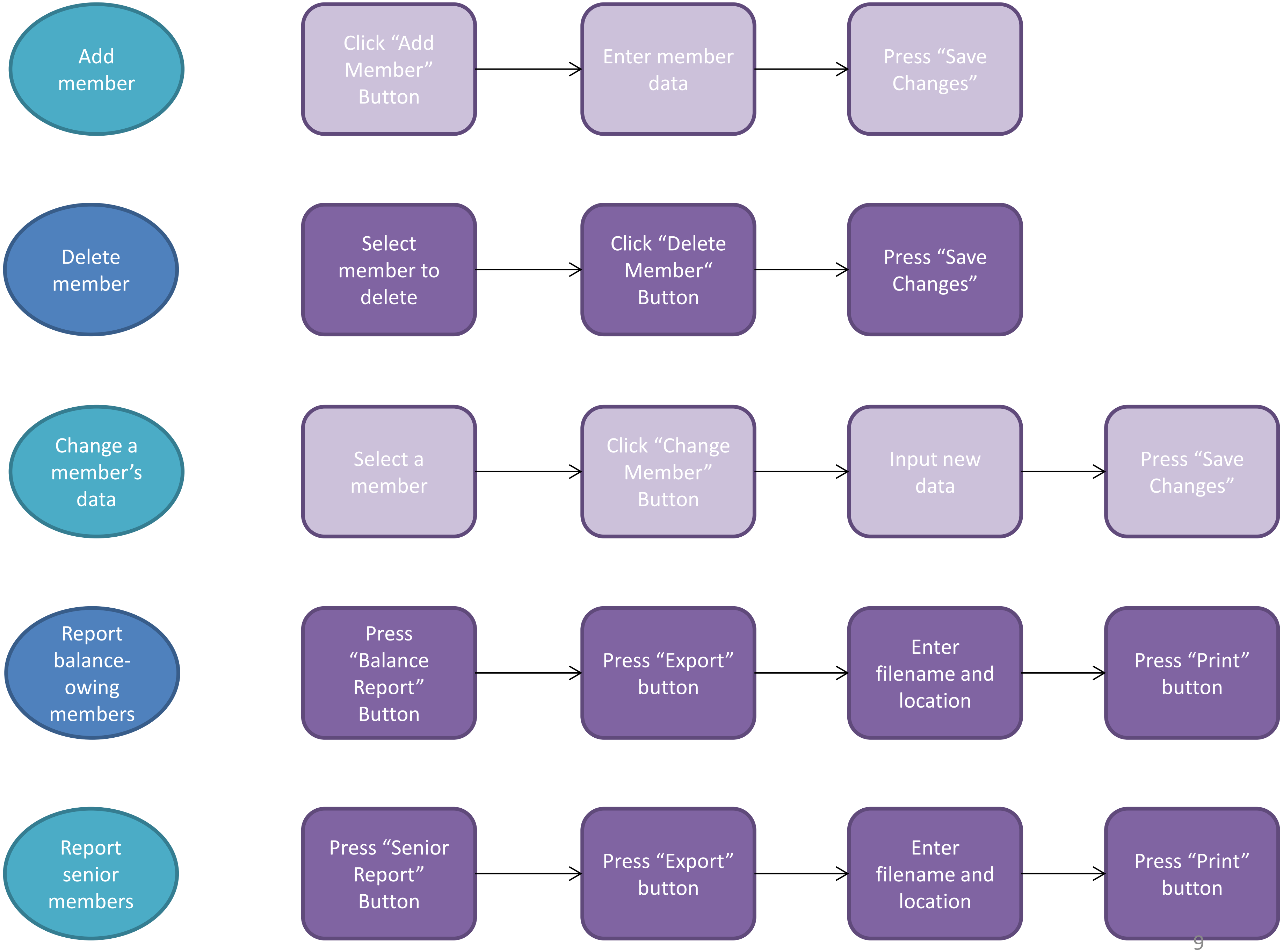
Add member

Delete member

Change a member's data

Report balance-owing members

Report senior members



Add member

Click “Add Member” Button → Enter member data → Press “Save Changes”

Delete member

Select member to delete → Click “Delete Member“ Button → Press “Save Changes”

Change a member’s data

Select a member → Click “Change Member” Button → Input new data → Press “Save Changes”

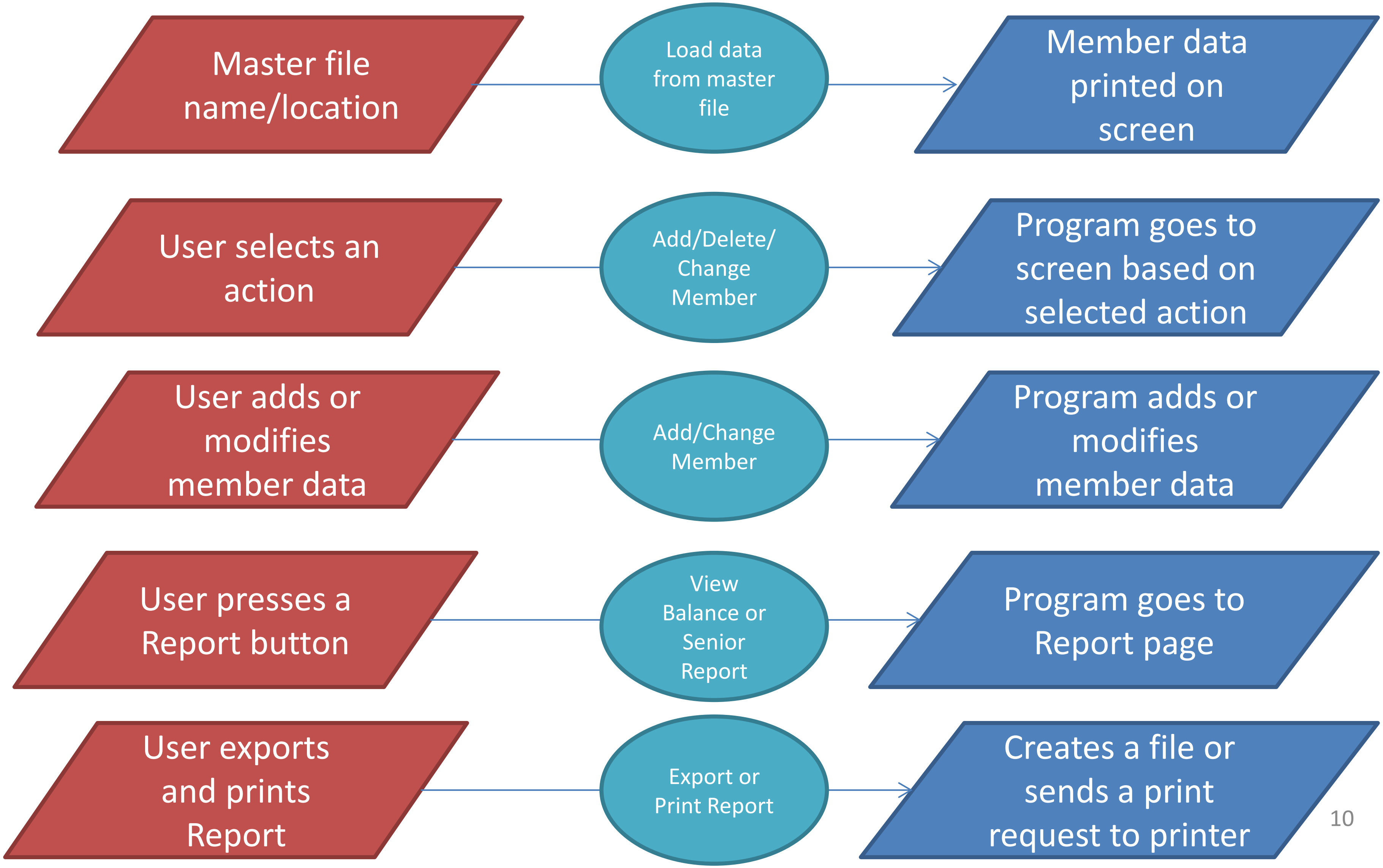
Report balance-owing members

Press “Balance Report” Button → Press “Export” button → Enter filename and location → Press “Print” button

Report senior members

Press “Senior Report” Button → Press “Export” button → Enter filename and location → Press “Print” button

Program Input/Output



Master file name/location

Load data from master file

Member data printed on screen

User selects an action

Add/Delete/ Change Member

Program goes to screen based on selected action

User adds or modifies member data

Add/Change Member

Program adds or modifies member data

User presses a Report button

View Balance or Senior Report

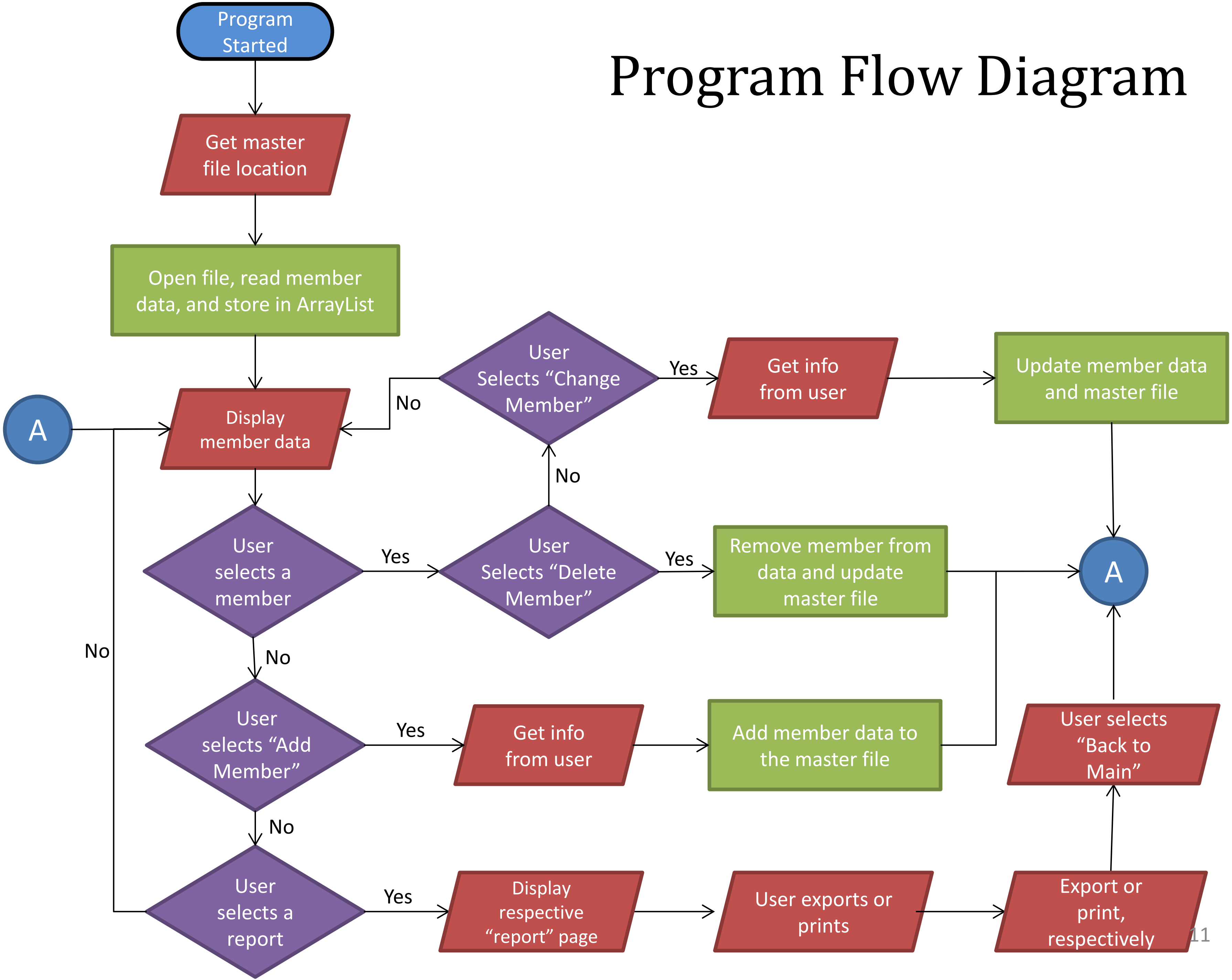
Program goes to Report page

User exports and prints Report

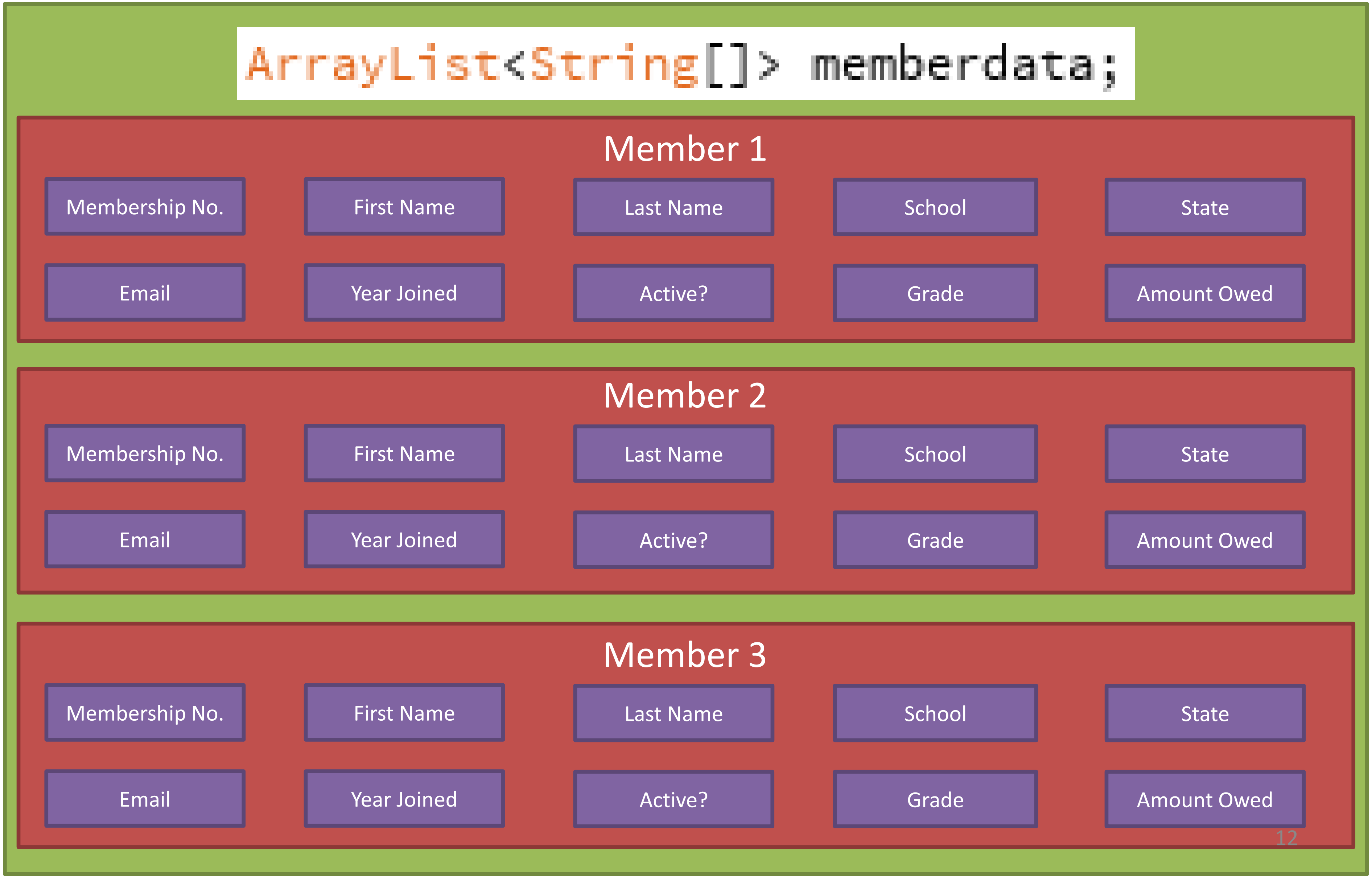
Export or Print Report

Creates a file or sends a print request to printer

Program Flow Diagram


Program Started
Get master file location
Open file, read member data, and store in ArrayList
A
Display member data
User Selects "Change Member"
Yes
Get info from user
Update member data and master file
No
No
User selects a member
Yes
User Selects "Delete Member"
Yes
Remove member from data and update master file
A
No
No
User selects "Add Member"
Yes
Get info from user
Add member data to the master file
User selects "Back to Main"
No
User selects a report
Yes
Display respective "report" page
User exports or prints
Export or print, respectively

Program Structure: Data Structure



```
ArrayList<String[]> memberdata;
```

Member 1

Membership No. | First Name | Last Name | School | State

Email | Year Joined | Active? | Grade | Amount Owed

Member 2

Membership No. | First Name | Last Name | School | State

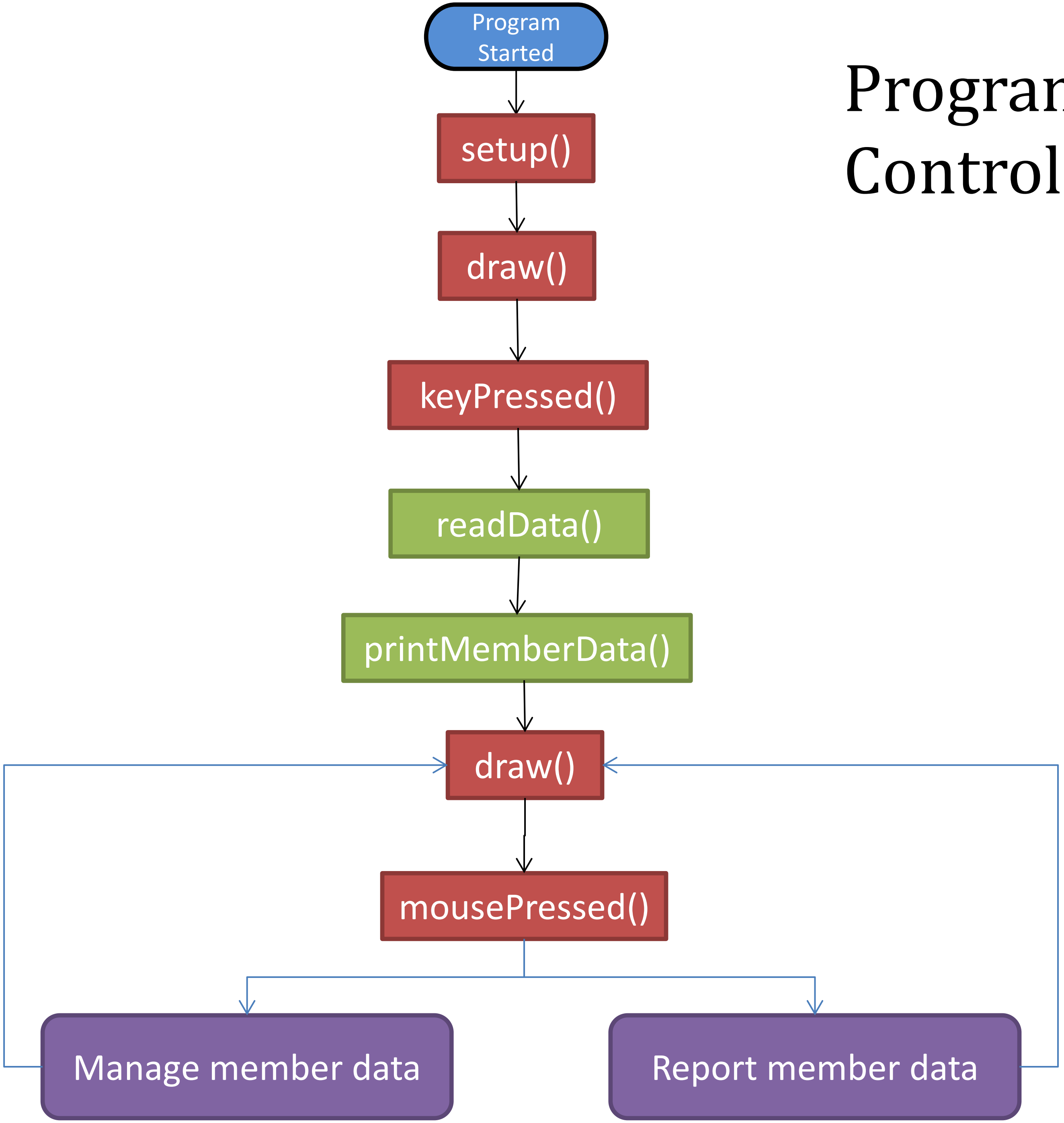
Email | Year Joined | Active? | Grade | Amount Owed

Member 3

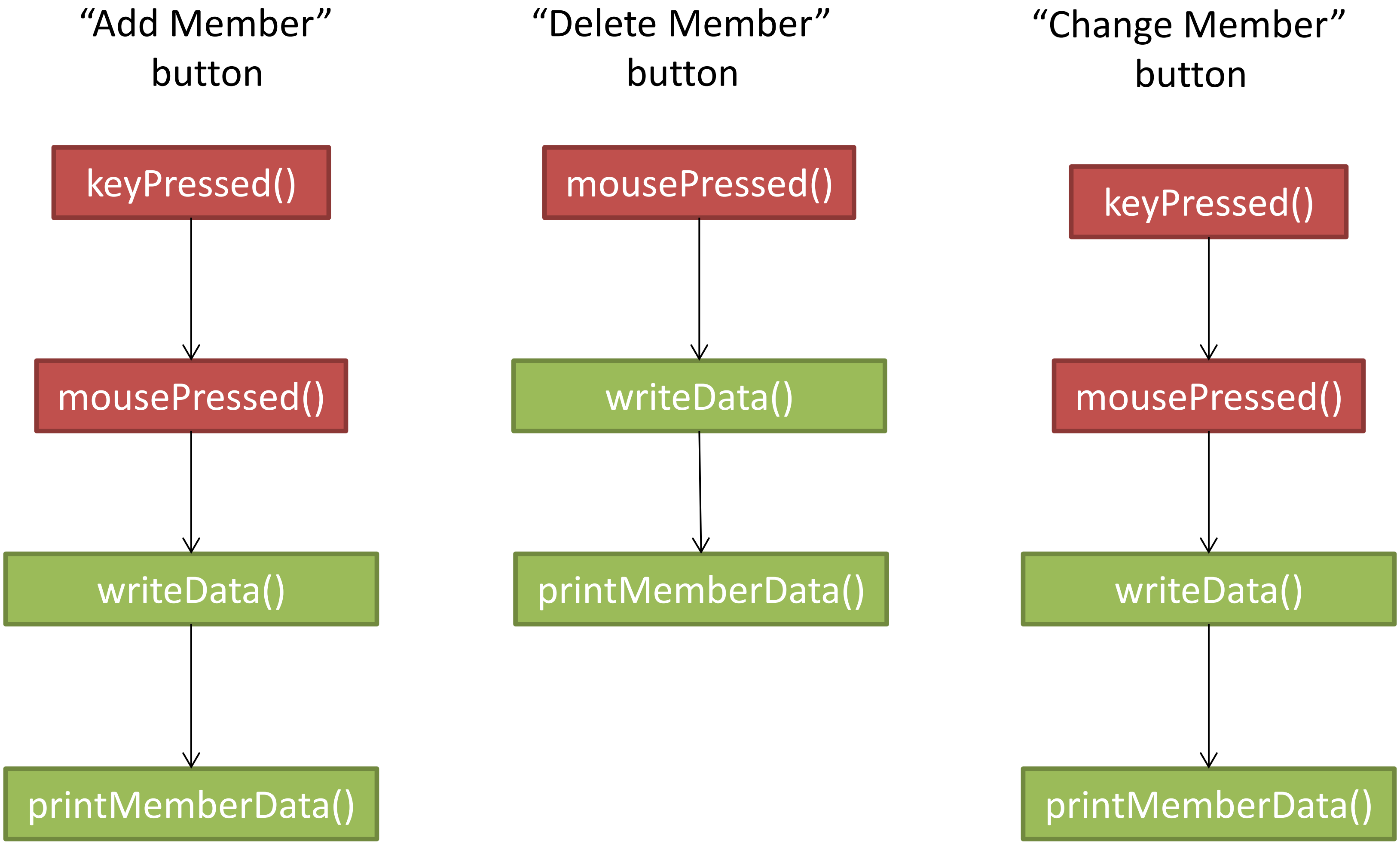
Membership No. | First Name | Last Name | School | State

Email | Year Joined | Active? | Grade | Amount Owed

Program Structure: Control Structure



Manage member data



“Add Member” button

keyPressed()

mousePressed()

writeData()

printMemberData()

“Delete Member” button

mousePressed()

writeData()

printMemberData()

“Change Member” button

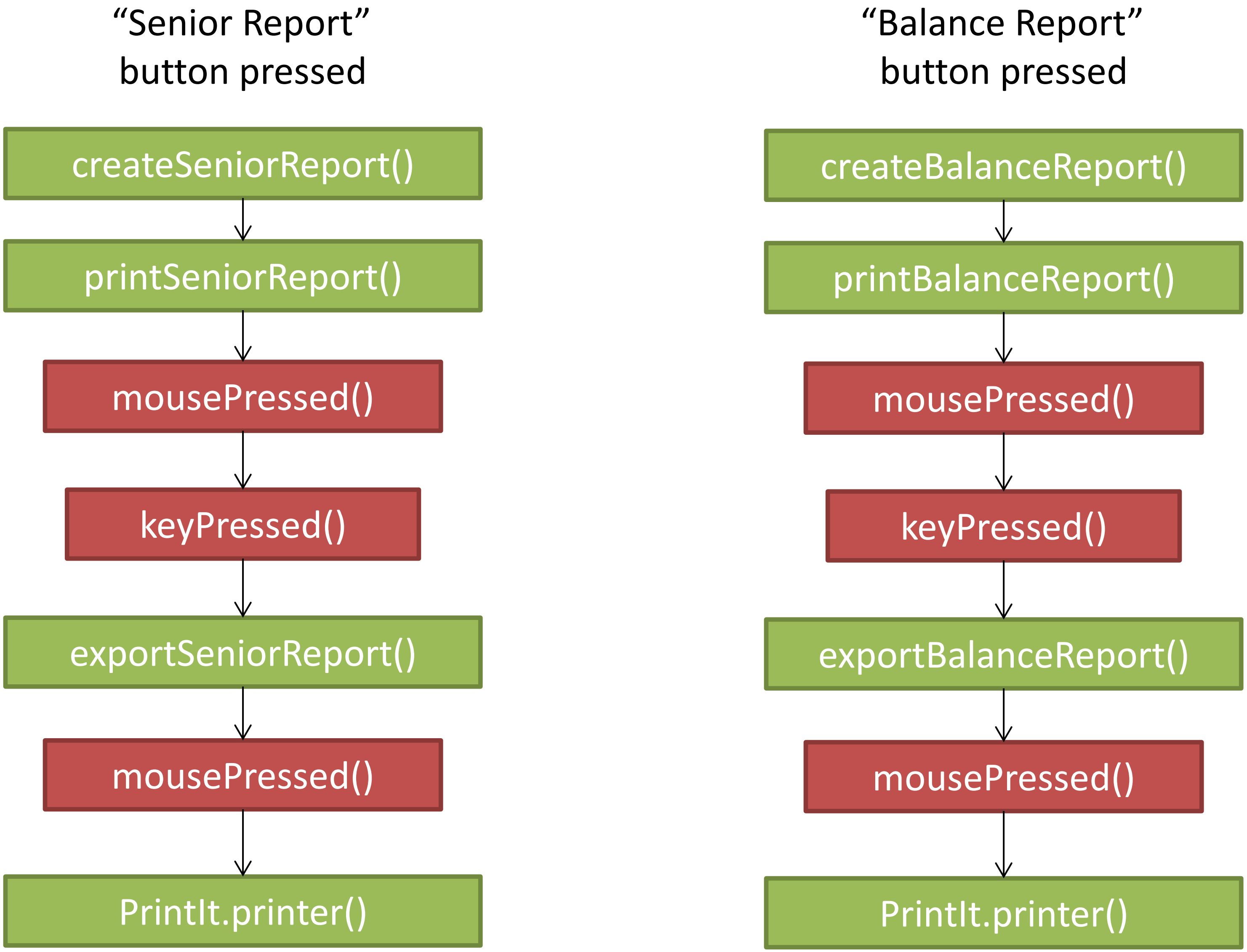
keyPressed()

mousePressed()

writeData()

printMemberData()

Report member data



“Senior Report” button pressed

createSeniorReport()

printSeniorReport()

mousePressed()

keyPressed()

exportSeniorReport()

mousePressed()

PrintIt.printer()

“Balance Report” button pressed

createBalanceReport()

printBalanceReport()

mousePressed()

keyPressed()

exportBalanceReport()

mousePressed()

PrintIt.printer()