**Project 2**

**<Event Planner>**

**CIS-17A 42448**

**Name: Andrew Kim**

**Date: 6/6/201**

**Introduction**

Title: Event Planner

The program uses structures to organize a planner for a month. The events are created to simulate a calendar which can be utilized. The user can use the program to create a monthly plan using menu navigation. The menu will display different events to which the user can input which day will have those events. There are many other functionality to the program which takes advantages of the class concept.

**Summary**

Project Size: about 1030 lines

Main source: 590 lines

Header and classes: 440 lines

The number of variables: 35+ variables

The number of methods: 10 methods

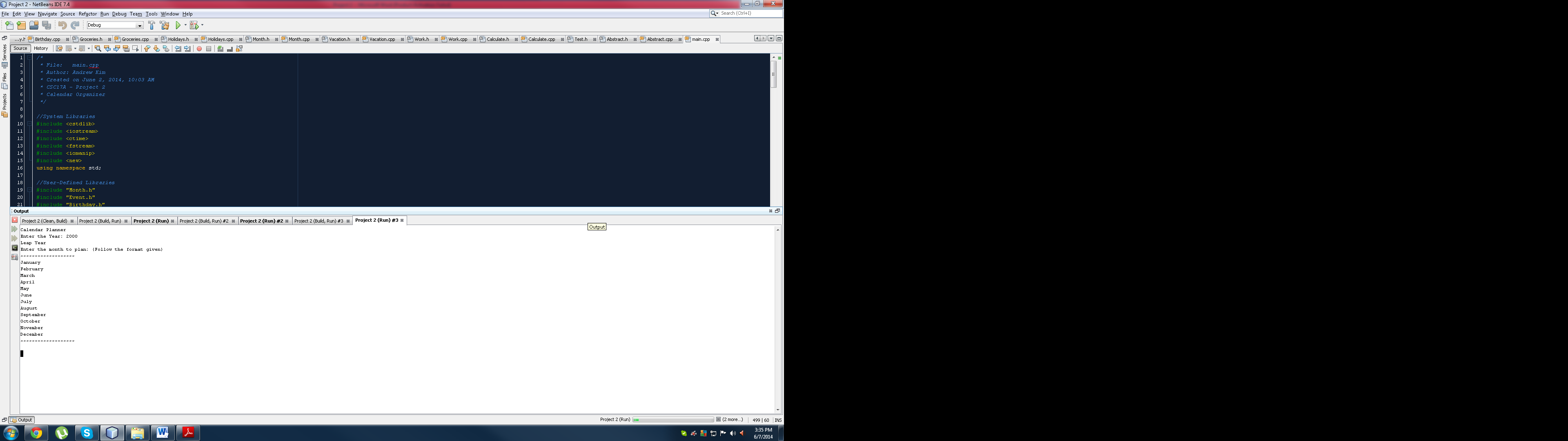
In the process of completing the project, abstract data type took a while to learn how to utilize. The amount of classes used in this project got hard to keep track, but when utilizing structures for planning the month and the event for the program was easy to implement. The classes made implementing different days of the month easy to figure out such as finding the day, finding the event by search and so on.

**Description**

The creation of the event planner was to use the class concept. Using the class concept made the program easier than if one were to create the program without it.

**Input / Output**

Most of the program is navigated using a menu system.



By looking at the figure above all one needs to do is follow the directions.

The menu says to enter the month to plan using the format given.

So the user could type: January

This will tell the program to make a monthly plan for January.

**Pseudocode**

*Initialize*

*Do while loop until program is done*

*Display Menu*

*Choose 1 : Enter the events for each day manually*

*Choose 2: Randomize the amount of events for each type*

*Choose 3: Reset all data*

*~all variables 0 or “”*

*Choose 4: Type in Event type to get the days corresponding to it*

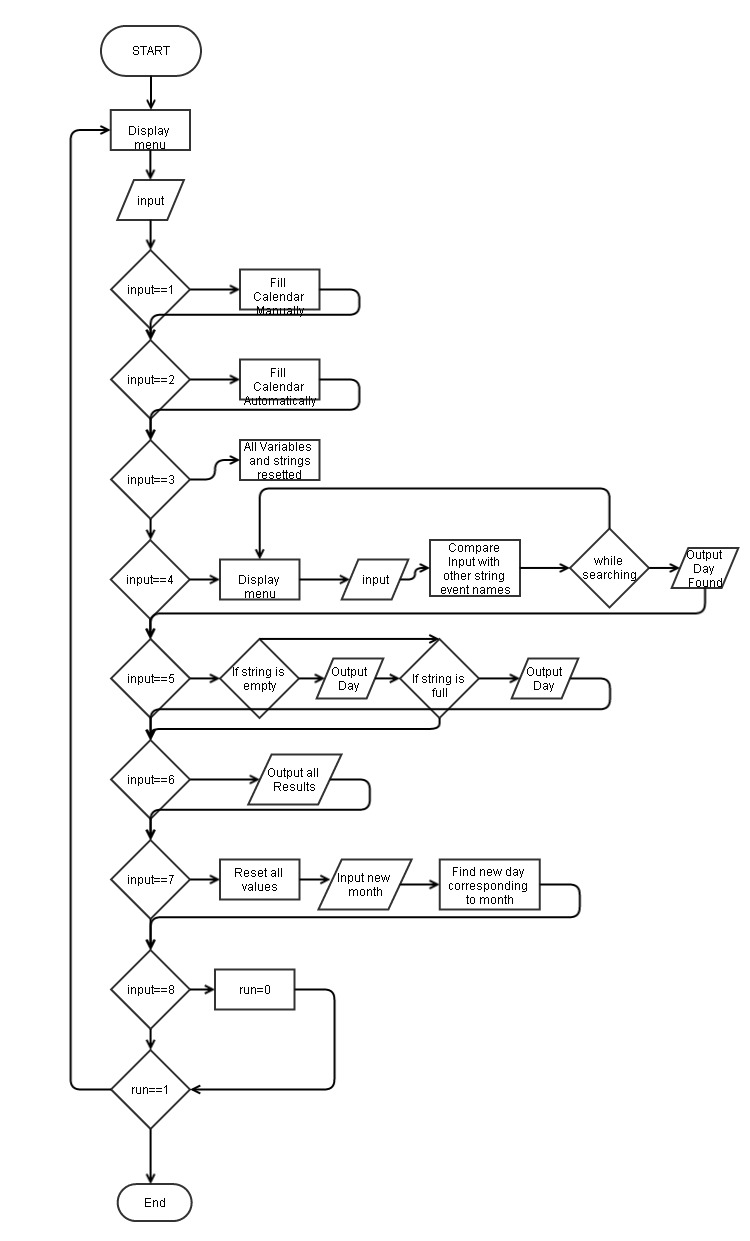
*Choose 5: Find days which are open or free*

*Choose 6: Display Result of everything*

*Choose 7: Change the month*

*Choose 8: Exit*

**Flowchart**

****

**Major Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variable Name** | **Description** | **Location** |
| integer | finish | variables to check to end program | main |
| integer | amount | stores amount for each event | main |
| integer | choice1 | Input for first menu | main |
| integer | choice2 | input for second menu | main |
| integer | choice3 | input for vacation operator menu | main |
| string | files | reads in file from text | main |
| integer | year | get year to see if there is a leapyear | main |
| string | trash | picks up string leftovers | main |

**Concepts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Concept** | **Variable Name** | **Description** | **Location** |
| CH9 pointers | ptr | used to carry percentage values | main |
| CH10 strings | files | carry string from file | main |
| CH10 strings | name | carry a name | main |
| CH10 strings | da[i] | string array to carry event names | main |
| CH11 structures | menu | use to strong a single string "January" | main |
| CH12 File | files | reads in from file | main |
| CH13 Classes | Every class header | used to organize data from a month | In all Header files |
| CH14 Class Operators | Vacation operator++() | function which counts up vacation days | vacation.h |
| CH15 Mulitiple Inheritance | Event | Used to find total event # | event.h |
| CH16 Exceptions | no variables | to check if there is bad allocation memory | line 496 on main |

**Reference**

Textbook

Power Point

In Class Examples

**Program**

/\*

\* File: main.cpp

\* Author: Andrew Kim

\* Created on June 2, 2014, 10:03 AM

\* CSC17A - Project 2

\* Calendar Organizer

\*/

//System Libraries

#include <cstdlib>

#include <iostream>

#include <ctime>

#include <fstream>

#include <iomanip>

#include <new>

using namespace std;

//User-Defined Libraries

#include "Month.h"

#include "Event.h"

#include "Birthday.h"

#include "Groceries.h"

#include "Holidays.h"

#include "Vacation.h"

#include "Work.h"

#include "Calculate.h"

#include "Abstract.h"

//Function Prototypes

void findDays(string &,int &,int);

//Execution Begins Here

int main(int argc, char\*\* argv) {

//Declare Variables

int days,finish=0,plan=0,choice1,choice2,choice3;

int full=0;

int year,leap;

int check1=0,check2=0;

//For Randomize Event for Calendar

srand(static\_cast<unsigned int>(time(0)));

//Variables being used on repeat

int amount[5]={0},day;

string name,trash;

//Use The classes in some way

Birthday evb(amount[0]);

Groceries evg(amount[1]);

Holidays evh(amount[2]);

Vacation evv(amount[3]);

Work evw(amount[4]);

//Open File

fstream dataFile("file.txt", ios::in);

string files;

//Enter the Month

if(dataFile){

getline(dataFile,files,'$');

cout<<files<<endl;

}

//Get the Year (For Leap Years)

cout<<"Enter the Year: ";

cin>>year;

//Calculate if Leap Year or Not

if(year%4==0){

cout<<"Leap Year"<<endl;

leap=1;

}

else{

cout<<"Non-Leap year"<<endl;

leap=0;

}

//Get the Month

cout<<"Enter the month to plan: (Follow the format given)"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<"January"<<endl;

cout<<"February"<<endl;

cout<<"March"<<endl;

cout<<"April"<<endl;

cout<<"May"<<endl;

cout<<"June"<<endl;

cout<<"July"<<endl;

cout<<"August"<<endl;

cout<<"September"<<endl;

cout<<"October"<<endl;

cout<<"November"<<endl;

cout<<"December"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<endl;

cin>>name;

getline(cin,trash);

//Find the days in the Month

findDays(name,days,leap);

//Construct the Month class

Month date(name,days);

//String array to carry event name

string da[date.getDays()];

do{

cout<<endl;

cout<<"~~Menu to add in Events~~~"<<endl;

cout<<"First choose one of the first 2 options"<<endl;

cout<<"1. Enter the Events"<<endl;

cout<<"2. Randomize Events for the Month"<<endl;

cout<<"3. Clear Data"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<"4. Search up Events to find out days"<<endl;

cout<<"5. Find days which are open"<<endl;

cout<<"6. Display Result of the Planner"<<endl;

cout<<"7. Change the month"<<endl;

cout<<"8. Press to exit"<<endl;

cin>>choice1;

if(choice1==1)check1=1;

//If went to choice 2 and coming back to choice 1

if(choice1==1&&check2==1){

cout<<"Resetting Data"<<endl;

//Reset everything

for(int i=0;i<date.getDays();i++){

da[i]="";

}

//Using pointers for the array to =0

\*(amount+0)=0;

\*(amount+1)=0;

\*(amount+2)=0;

\*(amount+3)=0;

\*(amount+4)=0;

\*(amount+5)=0;

//Reset from Class

evb.reset();

evg.reset();

evh.reset();

evv.reset();

evw.reset();

full=0;

}

//If the Calendar is filled up

if(choice1==1&&full==1){

cout<<"Sorry. Event Calendar is full"<<endl;

}

//If the Calendar isn't filled up

if(choice1==1&&full!=1){

do{

cout<<"~~Choose the Events to add into the Monthly Planner~~~"<<endl;

cout<<"1. Birthday Events"<<endl;

cout<<"2. Groceries"<<endl;

cout<<"3. Holidays"<<endl;

cout<<"4. Vacations"<<endl;

cout<<"5. Work-Related"<<endl;

cout<<"6. Back to main menu"<<endl;

cin>>choice2;

//Birthdays

if(choice2==1){

//Amount of Birthdays

cout<<"How many Birthday events is there?"<<endl;

cin>>amount[0];

evb.setNumb(amount[0]);

//Enter Day

for(int i=0;i<amount[0];i++){

cout<<"Which day is the Birthday event?"<<endl;

cin>>day;

//If the Day is filled do this

if(da[day-1]!=""){

do{

cout<<"Event for this day is already filled"<<endl;

cout<<"Please pick another day. ";

cin>>day;

}while(da[day-1]!="");

}

da[day-1]="Birthday Event";

}

}

//Groceries

if(choice2==2){

//Amount of Groceries

cout<<"How many times do you buy groceries in a month?"<<endl;

cin>>amount[1];

evg.setNumb(amount[1]);

for(int i=0;i<amount[1];i++){

//Enter Day

cout<<"Which day do you buy groceries?"<<endl;

cin>>day;

//If the Day is filled do this

if(da[day-1]!=""){

do{

cout<<"Event for this day is already filled"<<endl;

cout<<"Please pick another day. ";

cin>>day;

}while(da[day-1]!="");

}

da[day-1]="Groceries";

}

}

//Holiday

if(choice2==3){

//Amount of Holidays

cout<<"How many holidays are there this month?"<<endl;

cin>>amount[2];

evh.setNumb(amount[2]);

for(int i=0;i<amount[2];i++){

//Enter Day

cout<<"Which day is a holiday?"<<endl;

cin>>day;

//If the Day is filled do this

if(da[day-1]!=""){

do{

cout<<"Event for this day is already filled"<<endl;

cout<<"Please pick another day. ";

cin>>day;

}while(da[day-1]!="");

}

da[day-1]="Holiday";

}

}

//Vacation

if(choice2==4){

//Amount of Vacations

cout<<"How many vacations do you have this month?"<<endl;

cin>>amount[3];

cout<<"Do you want to add another vacation day?"<<endl;

cout<<"1. Yes"<<endl;

cout<<"2. No"<<endl;

cin>>choice3;

if(choice3==1){

evv.setNumb(amount[3]);

evv.operator ++();

amount[3]=amount[3]+evv.getAdd();

evv.setNuma(amount[3]);

}

if(choice3==2){

evv.setNumb(amount[3]);

}

for(int i=0;i<amount[3];i++){

//Enter Day

cout<<"Which day is your vacation?"<<endl;

cin>>day;

//If the Day is filled do this

if(da[day-1]!=""){

do{

cout<<"Event for this day is already filled"<<endl;

cout<<"Please pick another day. ";

cin>>day;

}while(da[day-1]!="");

}

da[day-1]="Vacation";

}

}

//Work

if(choice2==5){

//Amount of Work

cout<<"How many work related days you have this month?"<<endl;

cin>>amount[4];

evw.setNumb(amount[4]);

for(int i=0;i<amount[3];i++){

//Enter Day

cout<<"When do you have work related events?"<<endl;

cin>>day;

//If the Day is filled do this

if(da[day-1]!=""){

do{

cout<<"Event for this day is already filled"<<endl;

cout<<"Please pick another day. ";

cin>>day;

}while(da[day-1]!="");

}

da[day-1]="Work-Related Event";

}

}

if(choice2==6)plan=1;

}while(plan==0);

}

//To make it easier

int tot;

int dah=date.getDays();

//If already made options 1

if(choice1==2)check2=1;

//If made option 1 and choice option 2 also

if(choice1==2&&check1==1){

//Resets all data

cout<<"Resetting Data"<<endl;

//Reset everything

for(int i=0;i<date.getDays();i++){

da[i]="";

}

amount[0]=0;

amount[1]=0;

amount[2]=0;

amount[3]=0;

amount[4]=0;

amount[5]=0;

evb.reset();

evg.reset();

evh.reset();

evv.reset();

evw.reset();

full=0;

}

//If choice option 2 and Calendar is filled up

if(choice1==2&&full==1){

cout<<"Sorry. Event Calendar is full"<<endl;

}

//If choice option 2 and Calendar isn't filled up

if(choice1==2&&full!=1){

do{

//Check to see if there are more events than days in the month

cout<<"How many Birthdays in this month? :";

cin>>amount[0];

cout<<"How many times do you buy Groceries in a month: ";

cin>>amount[1];

cout<<"How many Holidays in this month: ";

cin>>amount[2];

cout<<"How many Vacation days do you have in this month: ";

cin>>amount[3];

cout<<"How many Work-related days do you have this month: ";

cin>>amount[4];

//Add the event amount

tot=amount[0]+amount[1]+amount[2]+amount[3]+amount[4];

if(tot>dah){

cout<<"There are more events then days in the month"<<endl;

cout<<"Re-Enter the amount"<<endl;

cout<<endl;

}

if(tot==dah)full=1;

//Exception

}while(tot>dah);

//Add Amount After the check

evb.setNumb(amount[0]);

evg.setNumb(amount[1]);

evh.setNumb(amount[2]);

evv.setNumb(amount[3]);

evw.setNumb(amount[4]);

//Randomize and put events into the days of the month

int a;

for(int i=0;i<amount[0];i++){

a=rand()%dah+1;

if(da[a-1]!=""){ //If not empty

do{

a=rand()%dah+1;

}while(da[a-1]!="");

}

da[a-1]="Birthday Event";

}

int b;

for(int i=0;i<amount[1];i++){

b=rand()%dah+1;

if(da[b-1]!=""){ //If not empty

do{

b=rand()%dah+1;

}while(da[b-1]!="");

}

da[b-1]="Groceries";

}

int c;

for(int i=0;i<amount[2];i++){

c=rand()%dah+1;

if(da[c-1]!=""){ //If not empty

do{

c=rand()%dah+1;

}while(da[c-1]!="");

}

da[c-1]="Holiday";

}

int d;

for(int i=0;i<amount[3];i++){

d=rand()%dah+1;

if(da[d-1]!=""){ //If not empty

do{

d=rand()%dah+1;

}while(da[d-1]!="");

}

da[d-1]="Vacation";

}

int e;

for(int i=0;i<amount[4];i++){

e=rand()%dah+1;

if(da[e-1]!=""){ //If not empty

do{

e=rand()%dah+1;

}while(da[e-1]!="");

}

da[e-1]="Work Related";

}

}

if(choice1==3){

cout<<"Resetting all data"<<endl;

//Reset everything

for(int i=0;i<date.getDays();i++){

da[i]="";

}

//Pointer

\*(amount+0)=0;

\*(amount+1)=0;

\*(amount+2)=0;

\*(amount+3)=0;

\*(amount+4)=0;

\*(amount+5)=0;

//Reset from Class

evb.reset();

evg.reset();

evh.reset();

evv.reset();

evw.reset();

full=0;

}

cout<<endl;

string search;

//Search for Events and get days with those events

if(choice1==4){

cout<<"Enter which event you want to find the days for."<<endl;

cout<<"Enter with the format of the following events below"<<endl;

cout<<"~Event~"<<endl;

cout<<"Birthday Event"<<endl;

cout<<"Groceries"<<endl;

cout<<"Holiday"<<endl;

cout<<"Vacation"<<endl;

cout<<"Work Related"<<endl;

cout<<endl;

cin.ignore();

getline(cin,search);

cout<<endl;

//If The format for search is wrong

if(search!="Birthday Event"&&search!="Groceries"&&search!="Holiday"&&search!="Vacation"&&search!="Work Related"){

do{

cout<<"The formating is wrong"<<endl;

cout<<"Please type in the Events in the correct format"<<endl;

cout<<"~Event~"<<endl;

cout<<"Birthday Event"<<endl;

cout<<"Groceries"<<endl;

cout<<"Holiday"<<endl;

cout<<"Vacation"<<endl;

cout<<"Work Related"<<endl;

cout<<endl;

getline(cin,search);

cout<<endl;

}while(search!="Birthday Event"&&search!="Groceries"&&search!="Holiday"&&search!="Vacation"&&search!="Work Related");

}

//Search Event in the Array

for(int i=0;i<date.getDays();i++){

if(search==da[i]){

cout<<"Found in Day "<<i+1<<endl;

}

}

cout<<endl;

}

if(choice1==5){

//Displays Free Days of the month

cout<<"The following days are free"<<endl;

for(int i=0;i<date.getDays();i++){

if(da[i]==""){

cout<<"Day "<<i+1<<endl;

}

}

cout<<endl;

//Displays Days with events in that month

cout<<"The following days are not free"<<endl;

for(int i=0;i<date.getDays();i++){

if(da[i]!=""){

cout<<"Day "<<i+1<<" is not free"<<endl;

}

}

cout<<endl;

}

cout<<endl;

if(choice1==6){

//Display Day # and Event corresponding to it

for(int i=0;i<date.getDays();i++){

cout<<date.getName()<<" "<<i+1<<": "<<da[i]<<endl;

}

cout<<endl;

//Displays the # for each event

cout<<"Birthday Events: "<<evb.getNub()<<endl;

cout<<"Groceries: "<<evg.getNug()<<endl;

cout<<"Holidays: "<<evh.getNuh()<<endl;

cout<<"Vacation: "<<evv.getNuv()<<endl;

cout<<"Work: "<<evw.getNuw()<<endl;

//Event Class to add up all Events happened

Event a(evb.getNub(),evg.getNug(),evh.getNuh(),evv.getNuv(),evw.getNuw());

//Adds all the events and displays the # for all events

a.addUp();

cout<<endl;

//Classes for Calculating Percentage

Calculate stuff(evb.getNub(),evg.getNug(),evh.getNuh(),evv.getNuv(),evw.getNuw());

//For Format and decimal places

cout<<setprecision(2)<<fixed<<endl;

//Use Dynamic Array

float \*ptr;

//Check with try catch

try{

ptr=new float[5];

}

catch(bad\_alloc){

cout<<"Insufficient memory"<<endl;

}

/\*

cout<<"In One Month"<<endl;

cout<<"Percentage of Birthdays: %"<<stuff.getBP()\*100<<endl;

cout<<"Percentage of Groceries: %"<<stuff.getGP()\*100<<endl;

cout<<"Percentage of Holidays: %"<<stuff.getHP()\*100<<endl;

cout<<"Percentage of Vacation: %"<<stuff.getVP()\*100<<endl;

cout<<"Percentage of Work: %"<<stuff.getWP()\*100<<endl;

\*/

//Shows percentage for each Event / Total Event

ptr[0]=stuff.getBP()\*100;

ptr[1]=stuff.getGP()\*100;

ptr[2]=stuff.getHP()\*100;

ptr[3]=stuff.getVP()\*100;

ptr[4]=stuff.getWP()\*100;

cout<<"Percentage in order (Birthday, Groceries, Holidays, Vacation, Work"<<endl;

for(int i=0;i<5;i++){

cout<<"%"<<ptr[i]<<" ";

}

cout<<endl;

cout<<"~~~Random Abstract Test~~~"<<endl;

Abstract abs("Person",5); //Person A wants 5 days of vacation

abs.seta(15);

cout<<"Gets: "<<abs.getStuff()<<endl;

//

//Delete

delete []ptr;

cout<<endl;

}

if(choice1==7){

//Reset everything

for(int i=0;i<date.getDays();i++){

da[i]="";

}

//Pointer to 0

\*(amount+0)=0;

\*(amount+1)=0;

\*(amount+2)=0;

\*(amount+3)=0;

\*(amount+4)=0;

\*(amount+5)=0;

//Reset from Class all to 0

evb.reset();

evg.reset();

evh.reset();

evv.reset();

evw.reset();

full=0;

//Changes the month by input

cout<<"Changing the Month"<<endl;

cout<<"Enter the month to plan: (Follow the format given)"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<"January"<<endl;

cout<<"February"<<endl;

cout<<"March"<<endl;

cout<<"April"<<endl;

cout<<"May"<<endl;

cout<<"June"<<endl;

cout<<"July"<<endl;

cout<<"August"<<endl;

cout<<"September"<<endl;

cout<<"October"<<endl;

cout<<"November"<<endl;

cout<<"December"<<endl;

cout<<"~~~~~~~~~~~~~~~~~~~"<<endl;

cout<<endl;

cin>>name;

getline(cin,trash);

//Find the days in the Month

findDays(name,days,leap);

//Utilize the classes mutator functions

date.setDays(days);

date.setName(name);

}

if(choice1==8)finish=1;

}while(finish==0);

//Close the file

dataFile.close();

return 0;

}

void findDays(string &month,int &days,int leap){

string a[12]={"January","February","March","April","May","June","July","August","September","October","November","December"};

//Find the Days for that month

if(month=="January"||month=="March"||month=="May"||month=="July"||month=="August"||month=="October"||month=="December")days=31;

if(month=="April"||month=="June"||month=="September"||month=="November")days=30;

//28 days in non leap years

if(month=="February"&&leap==0)days=28;

if(month=="February"&&leap==1)days=29;

//If Months are spelled wrong or in a different format

if(month!="January"&&month!="February"&&month!="March"&&month!="April"&&month!="May"&&month!="June"&&month!="July"&&month!="August"

&&month!="September"&&month!="October"&&month!="November"&&month!="December"){

do{

cout<<"Re-Enter Month (Ex. January, February, March)"<<endl;

cin>>month;

//Check if the months are spelled right, if not keep looping until it is

}while(month!="January"&&month!="February"&&month!="March"&&month!="April"&&month!="May"&&month!="June"&&month!="July"

&&month!="August"&&month!="September"&&month!="October"&&month!="November"&&month!="December");

//Find the Days for that month

if(month=="January"||month=="March"||month=="May"||month=="July"||month=="August"||month=="October"||month=="December")days=31;

if(month=="April"||month=="June"||month=="September"||month=="November")days=30;

if(month=="February")days=28;

}

}