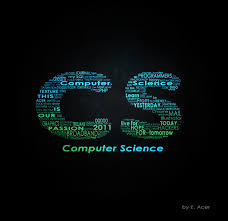
|  |
| --- |
|  |
| Computer Science |
| Internal Assessment |
|  |
| **Brandon Hinds** |
| **St. Jago High School** |

|  |
| --- |
| Done by: Brandon Hinds of form 6B3 |



**Table Of Contents**

Problem Statement and Solution ……………………………………….………………………………………….. page 1

Aims ………………………………………………………………………………………………………………………………. page 2

Background ………………………………………………………………………………………………………………….… page 3

Narration ……………………………………………………………………………………………………………………….. page 4

Pseudocode …………………………………………………………………………………………………….…….. pages 5 – 10

Test Plan ………………………………………………………………………………………………………………. pages 13 - 16

Program Code pages ………………………………………………………………………………………….…. pages 17 - 31

Problem Statement

At St. Jago High School the rental book system is hard to maintain. Having to go through records manually is time consuming and can be challenging. They need to keep records of the books they lend and the books that have been returned. They also need to find the books that they have that are in bad condition as well as the books that were returned in bad condition.

Solution  
What we could do is design a program to record the information of the rental books the school loans to students at the beginning of the year including their name, code, the condition they are in, who borrowed them and which class they are in and store all that information into a file. At the end of the year when the books are returned, they enter the books returned into the program and the program will store those books into another file. The user will be able to search for a book and will also be able to see which ones are in bad condition and those will be stored in a file.

AIMS

1. To allow the user to enter the name, code and condition of each book borrowed at the beginning of the year and store them in a text file.

2. To accept the name, code, condition and student info for each book Returned and store them in a text file.

3. To allow the user to search for a book previously entered in the list of borrowed books and display the information of the book.

4. To allow the user to search for a book previously entered in the list of Returned books and display the information of the book.

5. To display books in bad condition and store them in a text file. Background

St Jago High School (formerly Beckford & Smith) in Spanish Town, St Catherine, Jamaica, founded in 1744, is one of the oldest, continuously operated schools in the Western Hemisphere. It is renowned for graduating some of Jamaica's senior military officers, world class cricketers, academic scholars, performing artists, and Olympic athletes.

St jago is an offshoot of the Free School of Saint Jago de la Vega which started when Sir Peter Beckford in his will left 1,000 pound sterling to the construction of a free school, and Francis Smith left 300 pound sterling to the same cause. In 1958, after a merger of Beckford and Smith’s High and the Cathedral High, St. Jago High School was declared opened by then governor general of Jamaica, Sir Kenneth Blackburne.

St. Jago High school is located at Monk Street, Spanish Town, St. Catherine, Jamaica.

For years, St. Jago High School has given rental books to students of the school. Since the school has gotten bigger more rental books were obtained and issued. This growth has made it increasingly difficult to keep track of all books they have and rented. The school needs a program that will help them to manage the rental books.

Narration

When the program starts up a menu will appear informing the user of the options that are available. If the first option is chosen, the user should enter the name, code and condition of the book as well as the name of the student who borrowed it and the class they belong to. This information will be stored in a text file which will act as the database for all books that were loaned. If the second option is chosen the user should enter the name, code and condition of the book as well as the name of the student who borrowed it and the class they belong to. This information will be stored in another text file which will act as the database for all books returned. If the third option is chosen the user will have the chance to search for a book previously entered in one of the two files previously made. If the fourth option is chosen all books entered in the two files will be searched and the ones with the condition ‘bad’ will be printed onto the screen and into a new text file. If the fifth option is chosen the user will exit the program.

Pseudocode

Struct books {

READ Name;

READ Code;

READ Condition;

READ StudentName;

READ StudentClass;

}

Struct books library[4000]

FUNCTION MAIN:

MENU();

FUNCTION MENU:

Print ‘What Would You Like To Do?’;

Print ‘1. To Add Books To The File Containing Names Of Books Loaned’;

Print ‘2. To Add Books To The File Containing Names Of Books Returned’;

Print ‘3. To Search For A Previously Entered Book’

Print ‘4. To Find Books In Bad Condition’;

Print ‘5. To Leave Program’;

Read decision;

If (decision == 1):

ADDBOOK();

If (decision == 2):

Returned();

If (decision == 3):

SEARCHBOOK();

If (decision == 4):

CHECK();

If (decision == 5):

Exit();

FUNCTION ADDBOOK:

FILE(a) = Rental\_Books.txt;

Print ‘How many books do you have?’

Read NumOfBooks

Counter = 0;

While (Counter < NumOfBooks):

Print ‘Enter The Name of The Book Below’

Read books.Name

Print ‘Enter The Code of The Book Below’

Read books.Code

Print ‘Enter The Condition of The Book Below’

Read books.Condition

Print ‘Enter The Name Of The Student Who Borrowed The Book’

Read books.StudentName

Print ‘Enter The Class of The Student Below’

Read books.StudentClass

Print Name To File(a);

Print Code To File(a);

Print Condition To File(a);

Print books.StudentName To File(a);

Print books.StudentClass To File(a);

Close FILE(a);

FUNCTION Returned():

FILE(b) = Returned\_Books.txt;

Print ‘How many books do you have?’

Read NumOfBooks

Counter = 0;

While (Counter < NumOfBooks):

Print ‘Enter The Name of The Book Below’

Read books.Name

Print ‘Enter The Code of The Book Below’

Read books.Code

Print ‘Enter The Condition of The Book Below’

Read books.Condition

Print ‘Enter The Name Of The Student Who Borrowed The Book’

Read books.StudentName

Print ‘Enter The Class of The Student Below’

Read books.StudentClass

Print Name To File(b);

Print Code To File(b);

Print Condition To File(b);

Print books.StudentName To File(b);

Print books.StudentClass To File(b);

Close FILE(b);

FUNCTION SEARCHBOOK:

FILE(a) = Rental\_Books.txt;

Print ‘What Is The Name Of The Book?’

Read NSearch

While (Not End Of FILE(a)):

READ books.Name From FILE(a);

READ books.Code From File(a);

READ books.Condition From File(a);

READ books.StudentName From File(a);

READ books.StudentClass From File(a);

If (NSearch == book.Name):

Print books.Name;

Print books.Code;

Print books.Condition;

Print books.StudentName;

Print books.StudentClass;

Close FILE(a);

FUNCTION CHECK:

FILE(a) = Rental\_Books.txt;

File(b) = Rental\_Books.txt;

FILE(c) = Bad\_Books.txt

While (Not End Of FILE(c)):

READ books.Name From FILE(a);

READ books.Code From File(a);

READ books.Condition From File(a);

READ books.StudentName From File(a);

READ books.StudentClass From File(a);

READ books.Name From FILE(b);

READ books.Code From File(b);

READ books.Condition From File(b);

READ books.StudentName From File(b);

READ books.StudentClass From File(b);

If (books.Condition == ‘bad’):

Print books.Name To File(c);

Print books.Code To File(c);

Print books.Condition To File(c);

Print books.StudentName To File(c);

Print books.StudentClass To File(c);

Close FILE(a);

Close FILE(b);

Close FILE(c);

**Test Plan**

Test plan for menu function:

Five options should come up and the user should choose one of these options.

|  |  |  |
| --- | --- | --- |
| Input | Expected Results | Actual Results |
| 1 | User directed to the ‘AddBook’ function | The user was directed to the ‘AddBook’ function. |
| 2 | User directed to the ‘Returned’ function | The user was directed to the ‘Returned’ function. |
| 3 | User directed to ‘SearchBook’ function. | The user was directed to the ‘SearchBook’ function. |
| 4 | User directed to ‘Cheker’ function. | The user was directed to the ‘Checker’ function. |
| 5 | User gets to exit the program. | The program ended. |
| Anything other number. | An error statement should come up then the program should go back to main which will bring up the menu. | An error message came up and the user was redirected to the menu. |

Test plan for ‘AddBook’ function:

The user should input information for each book they loaned.

|  |  |  |
| --- | --- | --- |
| Input | Expected Results | Actual Results |
| Number of books the user wishes to enter. | The program should repeat the next five processes by the number just entered. | The program proceeded to enter the book information and this repeated according to the number just entered. |
| The name of the book. | It should be printed to the file containing the names of books loaned. | The user entered the name and it was stored to the file. |
| The code of the book. | It should be printed to the file containing the names of books loaned. | The user entered the code and it was stored to the file. |
| The condition of the book. | It should be printed to the file containing the names of books loaned. | The user entered the condition and it was stored to the file. |
| The name of the student who borrowed the book. | It should be printed to the file containing the names of books loaned. | The user entered the student’s name and it was stored to the file. |
| The class of the student who borrowed the book. | It should be printed to the file containing the names of books loaned. | The user entered the student’s class and it was stored to the file. |

Test plan for ‘Returned’ function:

The user should input information for each book returned.

|  |  |  |
| --- | --- | --- |
| Input | Expected Results | Actual Results |
| Number of books the user wishes to enter. | The program should repeat the next five processes by the number just entered. | The program proceeded to enter the book information and this repeated according to the number just entered. |
| The name of the book. | It should be printed to the file containing the names of books returned. | The user entered the name and it was stored to the file. |
| The code of the book. | It should be printed to the file containing the names of books returned. | The user entered the code and it was stored to the file. |
| The condition of the book. | It should be printed to the file containing the names of books returned. | The user entered the condition and it was stored to the file. |
| The name of the student who borrowed the book. | It should be printed to the file containing the names of books returned. | The user entered the student’s name and it was stored to the file. |
| The class of the student who borrowed the book. | It should be printed to the file containing the names of books returned. | The user entered the student’s class and it was stored to the file. |

Test plan for ‘SearchBook’ function:

The user should be able to search for a book previously entered.

|  |  |  |
| --- | --- | --- |
| Input | Expected Results | Actual Results |
| The name of the book. | The name, code, condition of the book as well as the name and class of the student who borrowed it should be printed onto the screen if the name of the book matches the name given. | The book with the name that matched the name that was just entered was printed along with the information that comes with it. |

Test plan for ‘Checker’ function:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The books with the condition ‘bad’ will be printed onto the screen and then into a file which contains names of all books in bad condition. | The book with the condition ‘bad’ were printed onto the screen along with the book information and all that was printed to the file. |

**Program Code**

#include <stdio.h>

#include <stdlib.h> //This library is used for the the exit function

#include <string.h> //This is used to compare string for the search function

typedef struct books{ //This struct is a model of the main information we want from each book

char code[10]; //The book code

char name[100]; //The book name

char condition[10]; //The book condition

char SName[100]; //The Name Of The Student who borrowed the book

char Class[50]; //The class of the student who borrowed the book

} Book;

//Function Declaration

int menu(); //This function is where the user will get to do whatever options are available

int AddBook(); //This function will allow the user to add books to the database

int Returned(); //This will allow the user to add the details of returned books to a file

int SearchBook(); //This function will allow the user to search for a specific book

int Checker(); //This function will print the books which are not in good conditions

//Global Variables

int n; //This is a counter and will also serve as the index for the array which holds the struct

int i; //This will be used to accept the number of books the user wants to enter

struct books library[4000]; //This array will hold the information of all books borrowed

struct books loaned[4000]; //This array will hold the information of books returned

Book book;

int main(){

menu();

return 0;

}

int menu(){

int choice; //This is the users choice of what to do first in this program

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome To St. Jago High's Rental Book System! \*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n");

printf("What Would You Like To Do?\n");

printf("Choose An Option By Selecting The Corresponding Number.\n");

printf("\n");

printf("1. To Add Books To The File Containing The List Of All Rental Books Loaned.\n");

printf("2. To Add Books To The File Containing The List Of Rental Books Returned.\n");

printf("3. To Search For A Book Previously Entered Into The List Of Books.\n");

printf("4. To Search For Books In Bad Or Unacceptable Condition.\n");

printf("5. To Exit This Program.\n");

scanf("%d",&choice);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

if (choice == 1){

AddBook(); //This will direct the user to function AddBook

}

else if (choice == 2){

Returned();

}

else if (choice == 3){

SearchBook(); //This will direct the user to function SearchBook

}

else if (choice == 4){

Checker(); //This will direct the user to the function Checker

}

else if (choice == 5){

exit(0); //The user can leave the program through this option

}

else{ //If the user enters anything than what is supposed to be entered:

printf("That Is Not An Option!\n");

return main(); //An error message will be printed and the program will return to main

}

}

int AddBook(){ //This is the function to add books to the file

FILE\*rental;

rental = fopen("Rental\_Books.txt","a");

printf("How Many Books Do You Want To Add?\n");

scanf("%d", &i);

for (n = 0; n < i; n++){

printf("Add The Names Of Rental Books Owned\n");

printf("\n");

printf("Enter The Book Name:\n");

scanf("%s",&library[n].name);

printf("Enter The Book Code:\n");

scanf("%s",&library[n].code);

printf("Enter The Book Condition (good or bad):\n");

scanf("%s",&library[n].condition);

printf("Enter The Name Of The Student Who Borrowed The Book:\n");

scanf("%s",&library[n].SName);

printf("Enter The Class Of The Student Who Borrowed The Book:\n");

scanf("%s",&library[n].Class);

printf("\n");

fprintf(rental, "%s\n" ,library[n].name);

fprintf(rental, "%s\n" ,library[n].code);

fprintf(rental, "%s\n" ,library[n].condition);

fprintf(rental, "%s\n" ,library[n].SName);

fprintf(rental, "%s\n" ,library[n].Class);

fprintf(rental, "\n");

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("The Information You Just Entered Has Been Stored To 'Rental\_Books.txt' File. \n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

fclose(rental);

return main(); //The program returns to main at the end of the function

}

int Returned(){

FILE\*borrow;

borrow = fopen("Returned\_Books.txt", "a");

printf("How Many Books That Have Been Returned Do You Want To Enter?\n");

scanf("%d", &i);

for (n = 0; n < i; n++){

printf("Enter The Book Name:\n");

scanf("%s",&loaned[n].name);

printf("Enter The Book Code:\n");

scanf("%s",&loaned[n].code);

printf("Enter The Book Condition (good or bad):\n");

scanf("%s",&loaned[n].condition);

printf("Enter The Name Of The Student Who Borrowed The Book:\n");

scanf("%s",&loaned[n].SName);

printf("Enter The Class Of The Student Who Borrowed The Book:\n");

scanf("%s",&loaned[n].Class);

printf("\n");

fprintf(borrow, "%s\n" ,loaned[n].name);

fprintf(borrow, "%s\n" ,loaned[n].code);

fprintf(borrow, "%s\n" ,loaned[n].condition);

fprintf(borrow, "%s\n" ,loaned[n].SName);

fprintf(borrow, "%s\n" ,loaned[n].Class);

fprintf(borrow, "\n");

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("The Information You Just Entered Has Been Stored To 'Returned\_Books.txt' File. \n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

fclose(borrow);

return main();

}

int SearchBook(){ //This is the function to search the file for a book previously entered

FILE\*rental;

FILE\*borrow;

borrow = fopen("Returned\_Books.txt", "r");

rental = fopen("Rental\_Books.txt","r");

char nsearch[100]; //This will be used to find the name of the book

if (rental == NULL){ //If the file is empty or does not exist:

printf("I'm Sorry But Information From The Rental\_Books Cannot Be Retrieved Right Now.\n");

return main(); //The error message will be printed and the program returns to main

}

else{

printf("The File 'Rental\_Books.txt' Will Be Searched\n");

printf("\n");

printf("What Is The Name Of The Book?\n");

scanf("%s", &nsearch);

for (n = 0; n < 4000; n++){

fscanf(rental, "%s" ,&library[n].name);

fscanf(rental, "%s" ,&library[n].code);

fscanf(rental, "%s" ,&library[n].condition);

fscanf(rental, "%s" ,&library[n].SName);

fscanf(rental, "%s" ,&library[n].Class);

if (borrow != NULL){

fscanf(borrow, "%s" ,&loaned[n].name);

fscanf(borrow, "%s" ,&loaned[n].code);

fscanf(borrow, "%s" ,&loaned[n].condition);

fscanf(borrow, "%s" ,&loaned[n].SName);

fscanf(borrow, "%s" ,&loaned[n].Class);

}

}

for (n = 0; n < 4000; n++){

if (strcmp(nsearch, library[n].name) == 0){

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Book Info no.%d (Borrowed)\n", n + 1);

printf("Name: %s\n", library[n].name);

printf("Code: %s\n", library[n].code);

printf("Condition: %s\n", library[n].condition);

printf("Student Name: %s\n" ,library[n].SName);

printf("Student Class: %s\n" ,library[n].Class);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

if (borrow != NULL){

if (strcmp(nsearch, loaned[n].name) == 0){

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Book Info no.%d (Returned)\n", n + 1);

printf("Name: %s\n", loaned[n].name);

printf("Code: %s\n", loaned[n].code);

printf("Condition: %s\n", loaned[n].condition);

printf("Student Name: %s\n" ,loaned[n].SName);

printf("Student Class: %s\n" ,loaned[n].Class);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

}

}

}

fclose(rental);

fclose(borrow);

return main(); //The program returns to main at the end of the function

}

int Checker() //This is function that will retrieve the information of the books under a certain condition

{

FILE\*rental;

FILE\*borrow;

FILE\*check;

rental = fopen("Rental\_Books.txt","r");

borrow = fopen("Returned\_Books.txt", "r");

check = fopen("Bad\_Books.txt","a");

char\*concheck = "bad";

if (rental == NULL){

printf("Information From The File Cannot Be Retrieved Right Now.\n");

return main();

}

else{

for (n = 0; n < 4000; n++){

fscanf(rental, "%s" ,&library[n].name);

fscanf(rental, "%s" ,&library[n].code);

fscanf(rental, "%s" ,&library[n].condition);

fscanf(rental, "%s" ,&library[n].SName);

fscanf(rental, "%s" ,&library[n].Class);

if (borrow != NULL){

fscanf(borrow, "%s" ,&loaned[n].name);

fscanf(borrow, "%s" ,&loaned[n].code);

fscanf(borrow, "%s" ,&loaned[n].condition);

fscanf(borrow, "%s" ,&loaned[n].SName);

fscanf(borrow, "%s" ,&loaned[n].Class);

}

}

for (n = 0; n < 4000; n++){

if (strcmp(concheck,library[n].condition) == 0){

printf("This Book Was In Bad Condition To Start With.\n");

printf("Name: %s\n", library[n].name);

printf("Code: %s\n", library[n].code);

printf("Condition: %s\n", library[n].condition);

printf("Student Name: %s\n" ,library[n].SName);

printf("Student Class: %s\n" ,library[n].Class);

printf("\n");

fprintf(check, "This Book Was In Bad Condition To Start With.\n");

fprintf(check, "Name: %s\n", library[n].name);

fprintf(check, "Code: %s\n", library[n].code);

fprintf(check, "Condition: %s\n", library[n].condition);

fprintf(check, "Student Name: %s\n" ,library[n].SName);

fprintf(check, "Student Class: %s\n" ,library[n].Class);

fprintf(check, "\n");

}

if (borrow != NULL){

if (strcmp(concheck,loaned[n].condition) == 0){

printf("This Book Was Returned In Bad Condition.\n");

printf("Name: %s\n", loaned[n].name);

printf("Code: %s\n", loaned[n].code);

printf("Condition: %s\n", loaned[n].condition);

printf("Student Name: %s\n" ,loaned[n].SName);

printf("Student Class: %s\n" ,loaned[n].Class);

printf("\n");

fprintf(check, "This Book Was Returned In Bad Condition.\n");

fprintf(check, "Name: %s\n", loaned[n].name);

fprintf(check, "Code: %s\n", loaned[n].code);

fprintf(check, "Condition: %s\n", loaned[n].condition);

fprintf(check, "Student Name: %s\n" ,loaned[n].SName);

fprintf(check, "Student Class: %s\n" ,loaned[n].Class);

fprintf(check, "\n");

}

}

}

}

fclose(rental);

fclose(borrow);

fclose(check);

return main();

}