

## **EXPERIMENT.NO-06**

**AIM:** Implementations of file handling operation using file stream such as create, open, read, write and close the file.

### **Objectives:-**

- 1) To understand and implement file handling concept.

### **Theory :-**

Input/Output with files

C++ provides the following classes to perform output and input of characters to/from files:

- ofstream: Stream class to write on files
- ifstream: Stream class to read from files
- fstream: Stream class to both read and write from/to files.

These classes are derived directly or indirectly from the classes istream, and ostream. We have already used objects whose types were these classes: cin is an object of class istream and cout is an object of class ostream. Therefore, we have already been using classes that are related to our file streams. And in fact, we can use our file streams the same way we are already used to use cin and cout, with the only difference that we have to associate these streams with physical files

Opening files using open()

The function open() can be used to open multiple files that use same stream

object . Ex: fstream inoutfile;

Inoutfile.open("book.dat");

Where filename is a null-terminated character sequence of type const char \* (the same type that string literals have) representing the name of the file to be opened, and mode is an optional parameter with a combination of the following flags:

ios::in Open for input operations.

ios::out Open for output operations.

ios::binary Open in binary mode.

ios::ate Set the initial position at the end of the file.

If this flag is not set to any value, the initial position is the beginning of the file.

`ios::app` All output operations are performed at the end of the file, appending the content to the current content of the file. This flag can only be used in streams open for output-only operations.

`ios::trunc` If the file opened for output operations already existed before, its previous content is deleted and replaced by the new one.

All these flags can be combined using the bitwise operator OR (`|`). For example, if we want to open the file `example.bin` in binary mode to add data we could do it by the following call to member function `open()`:

```
ofstream myfile;
```

```
myfile.open ("example.bin", ios::out | ios::app | ios::binary);
```

For `ifstream` and `ofstream` classes, `ios::in` and `ios::out` are automatically and respectively assumed, even if a mode that does not include them is passed as second argument to the `open()` member function.

The default value is only applied if the function is called without specifying any value for the mode parameter. If the function is called with any value in that parameter the default mode is overridden, not combined.

File streams opened in binary mode perform input and output operations independently of any format considerations. Non-binary files are known as text files, and some translations may occur due to formatting of some special characters (like newline and carriage return characters).

Since the first task that is performed on a file stream object is generally to open a file, these three classes include a constructor that automatically calls the `open()` member function and has the exact same parameters as this member. Therefore, we could also have declared the previous `myfile` object and conducted the same opening operation in our previous example by writing:

```
ofstream myfile ("example.bin", ios::out | ios::app | ios::binary);
```

Combining object construction and stream opening in a single statement. Both forms to open a file are valid and equivalent.

To check if a file stream was successful opening a file, you can do it by calling to member `is_open()` with no arguments. This member function returns a `bool` value of `true` in the case that indeed the stream object is associated with an open file, or `false` otherwise:

```
if (myfile.is_open()) { /* ok, proceed with output */ }
```

## **Closing a file**

When we are finished with our input and output operations on a file we shall close it so that its resources become available again. In order to do that we have to call the stream's

member function `close()`. This member function takes no parameters, and what it does is to flush the associated buffers and close the file:

```
myfile.close();
```

Once this member function is called, the stream object can be used to open another file, and the file is available again to be opened by other processes.

Checking state flags

`eof()` Returns true if a file open for reading has reached the end.

## Binary files

In binary files, to input and output data with the extraction and insertion operators (`<<` and `>>`) and functions like `getline` is not efficient, since we do not need to format any data, and data may not use the separation codes used by text files to separate elements (like space, newline, etc...).

File streams include two member functions specifically designed to input and output binary data sequentially: `write` and `read`. The first one (`write`) is a member function of `ostream` inherited by `ofstream`. And `read` is a member function of `istream` that is inherited by `ifstream`. Objects of class `fstream` have both members. Their prototypes are:

```
write ( memory_block, size );
```

```
read ( memory_block, size );
```

Where `memory_block` is of type "pointer to char" (`char*`), and represents the address of an array of bytes where the read data elements are stored or from where the data elements to be written are taken. The `size` parameter is an integer value that specifies the number of characters to be read or written from/to the memory block.

## Text files

Text file streams are those where we do not include the `ios::binary` flag in their opening mode. These files are designed to store text and thus all values that we input or output from/to them can suffer some formatting transformations, which do not necessarily correspond to their literal binary value.

Data output operations on text files are performed in the same way we operated with

`cout`: Functions for manipulation of file pointers :

All the actions on the file pointers take place automatically by default. The file stream classes support following functions to manage file pointers manually.

`tellg()` and `tellp()`

These two member functions have no parameters and return a value of the member type `pos_type`, which is an integer data type representing the current position of the get stream pointer (in the case of `tellg`) or the put stream pointer (in the case of `tellp`).

`seekg()` and `seekp()`

These functions allow us to change the position of the get and put stream pointers. Both functions are overloaded with two different prototypes. The first prototype is:

```
seekg ( position );
```

```
seekp ( position );
```

Using this prototype the stream pointer is changed to the absolute position position (counting from the beginning of the file). The type for this parameter is the same as the one returned by functions `tellg` and `tellp`: the member type `pos_type`, which is an integer value.

**// C++ program to create, write and close the file**

```
#include<iostream>
#include<fstream>
using namespace std;

int main()
{
    ofstream ofile;
    ofile.open ("abc.txt");
    ofile << "SYIT" << endl;
    cout << "GHRCEM" << endl;
    ofile.close();
    return 0;
}
```

**// C++ program to read the file**

```
#include<iostream>
#include<fstream>
using namespace std;

int main()
{
    char data[100];
    ifstream ifile;

    //create a text file before executing.
    ifile.open ("abc.txt");
    while ( !ifile.eof() )
    {
        ifile.getline (data, 100);
        cout << data << endl;
    }
}
```

```
ifile.close();  
return 0;  
}
```

### **Project details:**

In this Experiment =>

**A.** Two files are created (**windows.txt** and **macos.txt**) file created using file handling concepts for reading purpose.

And performed following operations

1. open file and read the data using ifstream.
- 2.. close the file using the myfile.close();

**B.** The **saved\_details.txt** file is created for the writing purpose i.e it add all the data selected by the user in the file.(if the file is already present the details will be saved to existing file).

And performed following operations

1. Write the data using ofstream
2. close the file using the myfile.close();

### **Code for File handling for our Project:**

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
#include<fstream>
```

```
#include<ctime>
```

```
#define MAX 5
```

```
using namespace std;
```

```
class User
```

```

{
public:
    string name;

    string lastname;

    string fn;
public:
    User()//constructor used
    {
        cout<<"\n * Enter the First name : ";

        cin>>name;

        cout<<"\n * Enter the lastname : ";

        cin>>lastname;
    }

    string fullname()
    {
        fn=name+" "+lastname;

        return fn;
    }

    void display()
    {
        cout<<"\n\n **Hello "<<fn<<" Welcome to our Program !!!"<<endl;
    }

    ~User()//Destructor called
    {
        cout<<"\n\n User Destructed Successfully !!!\n\n"<<endl;
    }
}

```

```
        cout.width(82);

        cout.fill('=');//formatted operation

        cout<<"\n\n";

    }

};
```

```
class Features:public User //Single Inheritance Class Feature Derived From Class User
{

    int os;

    int processor;

    int ram;

    int graphics_card;

    int memory;

    int ssd;

    int hdd;

    int both;//both is used for SSD and HDD

    int laptop_price;

public:

    void select_os();

    void select_ram();

    void select_processor();

    void select_graphics_card();

    void select_memory();

    void display_features();

    void select_laptop_price();
```

```

void filewrite();

void fileread();

};

void Features::filewrite()

{

    char fchoice;

    cout<<"\n \n Do You want to Save your Details ? y(Yes)/n(No)"<<endl;

    cout<<"\n Enter your Choice : "<<endl;

    cin>>fchoice;


    if(fchoice=='y'||'Y')

    {

        ofstream myfile;

        myfile.open("saved_details.txt",ios::out|ios::app);

        myfile<<"\n";


        myfile<<"\n Specifications Selected :";

        myfile<<"\n";

        myfile.width(30);

        myfile.fill('=');

        myfile<<"\n";

        myfile<<"\n Selected By : "<<fn<<endl;

        time_t tt;

        // Declaring variable to store return value of

```



```
// localtime()

struct tm * ti;

// Applying time()

time (&tt);

// Using localtime()

ti = localtime(&tt);

myfile<< "Selected on (Day ,Date & Time) = " << asctime(ti);


if(os==1){

myfile<<"\n OS: Windows";

if(processor==1)

myfile<<"\n Processor ==> Intel i3"<<endl;

else if(processor==2)

myfile<<"\n Processor ==> Intel i5"<<endl;

else if(processor==3)

myfile<<"\n Processor ==> Intel i7"<<endl;

else if(processor==4)

myfile<<"\n Processor ==> Intel i9"<<endl;

else if(processor==5)

myfile<<"\n Processor ==> Ryzen 5"<<endl;

else if(processor==6)

myfile<<"\n Processor ==> Ryzen 7"<<endl;


//RAM
```

```
if(ram==1)

    myfile<<"\n RAM ==> 4 GB"<<endl;

else if(ram==2)

    myfile<<"\n RAM ==> 8 GB"<<endl;

else if(ram==3)

    myfile<<"\n RAM ==> 16 GB"<<endl;

else if(ram==4)

    myfile<<"\n RAM ==> 32 GB"<<endl;


//graphics card

if(graphics_card==1)

    myfile<<"\n Graphics Card ==> 2 GB"<<endl;

else if(graphics_card==2)

    myfile<<"\n Graphics Card ==> 4 GB"<<endl;

else if(graphics_card==3)

    myfile<<"\n Graphics Card ==> 6 GB"<<endl;

else if(graphics_card==4)

    myfile<<"\n Graphics Card ==> 8 GB"<<endl;

else if(graphics_card==5)

    myfile<<"\n Integrated Graphics Card"<<endl;

else if(graphics_card==6)

    myfile<<"\n No Graphics card Selected."<<endl;


//Memory

if(memory==1)
```

```
    myfile<<"\n Memory Type ==> SSD"<<endl;
else if(memory==2)
    myfile<<"\n Memory Type ==> HDD"<<endl;
else if(memory==3)
    myfile<<"\n Memory Type ==> Both SSD & HDD"<<endl;
```

```
//SSD
```

```
if(ssd==1)
    myfile<<"\n SSD Capacity ==> 128 GB"<<endl;
else if(ssd==2)
    myfile<<"\n SSD Capacity ==> 256 GB"<<endl;
else if(ssd==3)
    myfile<<"\n SSD Capacity ==> 512 GB"<<endl;
else if(ssd==4)
    myfile<<"\n SSD Capacity ==> 1 TB"<<endl;
```

```
//HDD
```

```
if(hdd==1)
    myfile<<"\n HDD capacity ==> 128 GB"<<endl;
else if(hdd==2)
    myfile<<"\n HDD capacity ==> 256 GB"<<endl;
else if(hdd==3)
    myfile<<"\n HDD capacity ==> 512 GB"<<endl;
else if(hdd==4)
    myfile<<"\n HDD capacity ==> 1 TB"<<endl;
```

```
//BOTH
```

```
if(both==1)
```

```
    myfile<<"\n Memory ==> 128 GB SSD , 1 TB HDD"<<endl;
```

```
else if(both==2)
```

```
    myfile<<"\n Memory ==> 256 GB SSD , 1 TB HDD"<<endl;
```

```
else if(both==3)
```

```
    myfile<<"\n Memory ==> 512 GB SSD , 1 TB HDD"<<endl;
```

```
//price range
```

```
if(laptop_price==1)
```

```
    myfile<<"\n Price Range : 20,000 and Below"<<endl;
```

```
else if(laptop_price==2)
```

```
    myfile<<"\n Price Range : 20,000 - 30,000"<<endl;
```

```
else if(laptop_price==3)
```

```
    myfile<<"\n Price Range : 30,000 - 40,000"<<endl;
```

```
else if(laptop_price==4)
```

```
    myfile<<"\n Price Range : 40,000 - 50,000"<<endl;
```

```
else if(laptop_price==5)
```

```
    myfile<<"\n Price Range : 50,000 - 60,000"<<endl;
```

```
else if(laptop_price==6)
```

```
    myfile<<"\n Price Range : 60,000 - 70,000"<<endl;
```

```
else if(laptop_price==7)
```

```
    myfile<<"\n Price Range : 70,000 and Above\n\n"<<endl;
```

```
    myfile.close();
```

```
}

else if(os==2){

    myfile<<"\n OS : MAC";

//processor mac

if(processor==1)

    myfile<<"\n Processor ==> Intel i3"<<endl;

else if(processor==2)

    myfile<<"\n Processor ==> Intel i5"<<endl;

else if(processor==3)

    myfile<<"\n Processor ==> Intel i7"<<endl;

else if(processor==4)

    myfile<<"\n Processor ==> Intel i9"<<endl;

//ram mac

if(ram==1)

    myfile<<"\n RAM ==> 8 GB"<<endl;

else if(ram==2)

    myfile<<"\n RAM ==> 16 GB"<<endl;

// graphics card

if(graphics_card==1)

    myfile<<"\n Graphics Card ==> 4 GB"<<endl;

//memory

if(memory==1)

    myfile<<"\n Memory Type ==> SSD"<<endl;

//SSD
```

```

if(ssd==1)

    myfile<<"\n SSD Capacity ==> 256 GB"<<endl;

else if(ssd==2)

    myfile<<"\n SSD Capacity ==> 512 GB"<<endl;

else if(ssd==3)

    myfile<<"\n SSD Capacity ==> 1 TB"<<endl;

//price

if(laptop_price==1)

    myfile<<"\n Price Range : Rs. 90,000 and above "<<endl;

    myfile.close();

}

}

else if(fchoice=='n' || 'N')

{

    exit(0);

}

}

void Features::fileread()

{

    string data;

    ifstream myFile;

    int rchoice;

    cout<<"\n Do you want to see some Laptops ==>"<<endl;

    cout<<"\n 1.Windows"<<endl;

```

```
cout<<"\n 2.Apple"<<endl;

cout<<"\n 3.No(Exit)"<<endl;

cout<<"\n Enter your Choice : "<<endl;

cin>>rchoice;

switch(rchoice)

{

case 1:

    myFile.open("windows.txt");

    myFile>>data;

    while(!myFile.eof())

    {

        getline(myFile,data);

        cout<<"\n";

        cout<<data;

    }

    myFile.close();

    break;

case 2:

    myFile.open("macos.txt");

    myFile>>data;

    while(!myFile.eof())

    {

        getline(myFile,data);

        cout<<"\n";

        cout<<data;
```

```
        }

        myFile.close();

    case 3:

        break;

    default:

        cout<<"\n Enter valid choice !!"<<endl;

    }

}

class Brands {

char b_name[10];

public:

void getdata();

void b_display();

};

void Brands::getdata()

{

    cin>>b_name;

}

void Brands::b_display()

{

    cout<<"\n\n Brands Selected : \n"<<endl;

    cout<<b_name;

}

// Main Function

int main()
```



```

{

    int n,i;

    cout.width(82);

    cout.fill('=');//formatted operation

    cout<<"\n\n";

    cout<<"\t\t*****"<<endl;

    cout<<"\t\t*   Laptop Customization System   *"<<endl;

    cout<<"\t\t*****"<<endl;

    Features f;

    f.fullname();

    f.display();

    f.select_os();

    f.select_ram();

    f.select_processor();

    f.select_graphics_card();

    f.select_memory();

    f.select_laptop_price();


    Brands b[MAX];

    cout<<"\n * Enter the Number of Brands : "<<endl;

    cin>>n;

    for(i=1;i<=n;i++)

    {

        cout << "\n\n Enter Name of Brand : " << i << ":\n";

        b[i].getdata();
    }
}

```

```

    }

    for(i=1;i<=n;i++)

    {

        b[i].b_display();

    }


f.display_features();

f.filewrite();

f.fileread();

cout<<"\n\n\n\n * Thank you Visit Again :) !!!! "<<endl;

cout<<"\n * Developed By ==>"<<endl;

    cout<<"\n\n\n * Chris(SITA-06) \n\n * Varun(SITA-55) \n\n * Vipul(SITA-57) \n\n *
Rutwik(SITA-61) \n\n * Piyush(SITA-62)\n"<<endl;

    cout<<"\n  S.Y Information Technology Department !!!!"<<endl;

    cout<<"\n  G.H.Raisoni College of Engineering and Management,Wagholi,Pune .";

    cout<<"\n\n";

    return 0;


}

```

### Output:

```
=====
```

```
*****
```

```
* Laptop Customization System *
```

\*\*\*\*\*

\* Enter the First name : Piyush

\* Enter the lastname : Sonawane

\*\*Hello Piyush Sonawane Welcome to our Program !!!!

\*\* Please Select The Operating system you want in your Laptop :

1.Windows

2.MAC Os

3.Exit

Enter Your Choice : 2

\* You have Selected Your Os ==> MAC Os

=====

\*\*Select the RAM :

1. 8 GB

2. 16 GB

3. Exit

Enter Your Choice : 1

\* You have Selected Your RAM ==> 8 GB

=====

\*\*Select the Processor :

1. Intel i3

2. Intel i5

3. Intel i7

4. Intel i9

5. Exit

Enter Your Choice : 1

\* You have Selected Your Processor ==> Intel i3

=====

\*\*Select the Graphics card :

1. 4 GB

2. Don't Want graphics Card

3. Exit

Enter Your Choice : 1

\* You have Selected Your Graphics card ==> 4 GB

=====

What do you Want in Your Memory ?

1.SSD

2.Exit

Enter your Choice : 1

1.256 GB

2.512 GB

3.1 TB

4.Exit

Enter your Choice : 1

\* You have selected 256 GB SSD

\*\*Select the Price Range :

1. Rs. 90,000 - 1,00,000

2. Exit

Enter Your Choice : 1

Price Range selected Between : Rs. 90,000 and Above

\* Enter the Number of Brands :

1

Enter Name of Brand : 1:

Apple

Brands Selected :

Apple

\* Your Selected Specifications are ==>

Operating System ==> MAC OS

Processor ==> Intel i3

RAM ==> 8 GB

Graphics Card ==> 4 GB

Memory Type ==> SSD

SSD Capacity ==> 256 GB

Price Range : Rs. 90,000 and above

Do You want to Save your Details ? y(Yes)/n(No)

Enter your Choice :

y

Do you want to see some Laptops ==>

1.Windows

2.Apple

3.No(Exit)

Enter your Choice :

2

Some Suggested Laptops ==>

\*Apple MacBook Pro Core i3 10th Gen (8 GB/256 GB SSD)

(13.3 inch, Space Grey, 1.29 kg)

Ratings - 4.7/5.0

Rs. 92,990

\*Apple MacBook Pro Core i5 8th Gen (8 GB/256 GB SSD)

(13.3 inch, Space Grey, 1.37 kg)

Ratings - 4.6/5.0

Rs. 1,54,990

\*Apple MacBook Pro Core i7 9th Gen (16 GB/512 GB SSD)

(16 inch, Space Grey, 2 kg)

Ratings - 4.7/5.0

Rs. 1,99,900"

\* Apple MacBook Pro Corei9 8th Gen (16 GB/512 GB SSD /4 GB Graphics)

(15.4 inch, Space Grey, 1.83 kg)

Rating 4.1/5.0

Rs. 2,09,990"

\*Apple MacBook Pro Core i7 9th Gen (16 GB/512 GB SSD)

(16 inch, Space Grey, 2 kg)

Ratings - 4.7/5.0

Rs. 1,99,900

\*Apple MacBook Pro Core i5 10th Gen (16 GB/512 GB SSD)

(13 inch, Space Grey, 1.4 kg)

Ratings - 4.9/5.0

Rs. 1,74,900

\*Apple MacBook Pro Core i5 8th Gen (8 GB/512 GB SSD)

(13.3 inch, Space Grey, 1.4 kg)

Ratings - 4.7/5.0

Rs. 1,69,900

\*Apple MacBook Pro Core i9 9th Gen (16 GB/1 TB SSD/4 GB Graphics)

(16 inch, Space Grey, 2 kg)

Ratings - 4.7/5.0

Rs. 2,24,900

\* Apple MacBook Pro Corei7 8th Gen (16 GB/512 GB SSD /4 GB Graphics)"<<endl;

(15.4 inch, Space Grey, 1.83 kg)

Rating 4/5.0

Rs. 2,39,900

\* Apple MacBook Pro Corei5 10th Gen (16 GB/1 TB SSD )

(13 inch, Space Grey, 1.4 kg)

Rating 4.9/5.0

Rs. 1,94,900

\* Thank you Visit Again :) !!!!

\* Developed By ==>

\* Chris(SITA-06)

\* Varun(SITA-55)

\* Vipul(SITA-57)

\* Rutwik(SITA-61)

\* Piyush(SITA-62)

S.Y Information Technology Department !!!!

G.H.Raisoni College of Engineering and Management,Wagholi,Pune .

User Destructed Successfully !!!!

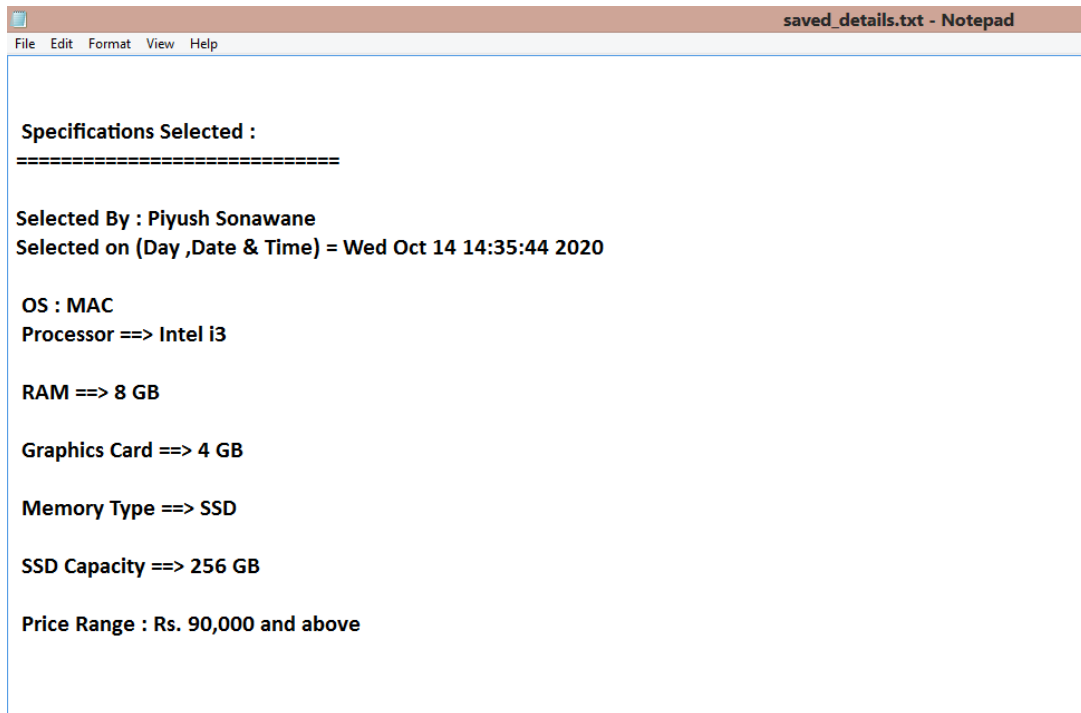
=====



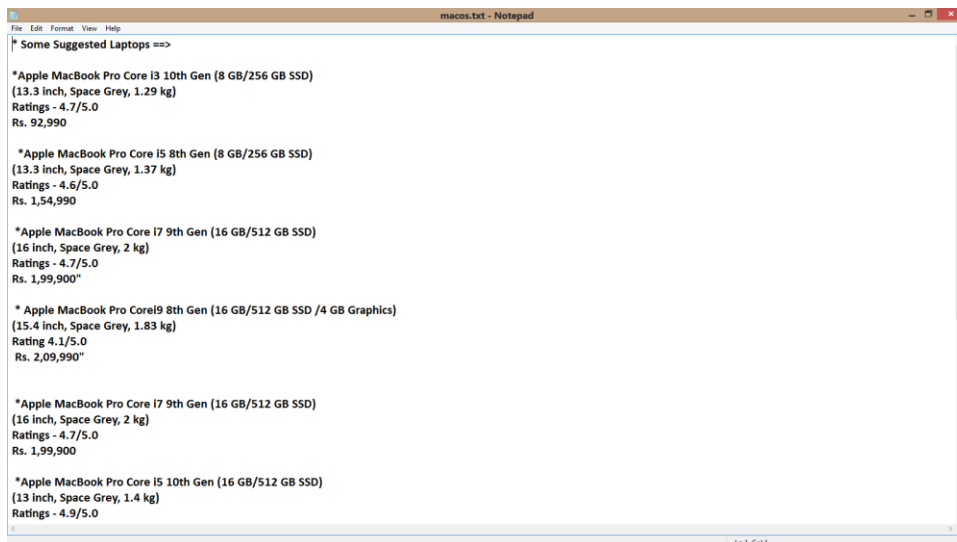
Process returned 0 (0x0) execution time : 59.661 s

Press any key to continue.

- **User entered Details are saved with user name and Date time of selection.**



- **If user want to see(read) the laptops so file windows.txt and macos.txt are there with data to read or import the data to program.**



## Output:

```
D:\c_cpp_programs\projetcpp.exe

*****
* Laptop Customization System *
*****

* Enter the First name : Piyush
* Enter the lastname : Sonawane

**Hello Piyush Sonawane Welcome to our Program !!!!
** Please Select The Operating system you want in your Laptop :

1.Windows
2.MAC Os
3.Exit
Enter Your Choice : 2

* You have Selected Your Os ==> MAC Os
=====

**Select the RAM :
1. 8 GB
2. 16 GB
3. Exit
Enter Your Choice : 1

* You have Selected Your RAM ==> 8 GB
=====

**Select the Processor :
1. Intel i3
2. Intel i5
3. Intel i7
4. Intel i9
5. Exit
Enter Your Choice : 1

* You have Selected Your Processor ==> Intel i3
=====

**Select the Graphics card :
1. 4 GB
2. Don't Want graphics Card
3. Exit
Enter Your Choice : 1

* You have Selected Your Graphics card ==> 4 GB
=====

What do you Want in Your Memory ?

1.SSD
2.Exit
Enter your Choice : 1
```

```
D:\c_cpp_programs\projetcpp.exe

2.Exit
Enter your Choice : 1
1.256 GB
2.512 GB
3.1 TB
4.Exit
Enter your Choice : 1
* You have selected 256 GB SSD

**Select the Price Range :
1. Rs. 90,000 - 1,00,000
2. Exit
Enter Your Choice : 1
Price Range selected Between : Rs. 90,000 and Above
* Enter the Number of Brands :
1
Enter Name of Brand : 1:
Apple

Brands Selected :
Apple

* Your Selected Specifications are ==>

Operating System ==> MAC OS
Processor ==> Intel i3
RAM ==> 8 GB
Graphics Card ==> 4 GB
Memory Type ==> SSD
SSD Capacity ==> 256 GB
Price Range : Rs. 90,000 and above

Do You want to Save your Details ? y<Yes>/n<No>
Enter your Choice :
y
Do you want to see some Laptops ==>
1.Windows
2.Apple
3.No<Exit>
Enter your Choice :
2
Some Suggested Laptops ==>
*Apple MacBook Pro Core i3 10th Gen <8 GB/256 GB SSD>
```

```
D:\c_cpp_programs\projetcpp.exe

Some Suggested Laptops ==>

*Apple MacBook Pro Core i3 10th Gen (8 GB/256 GB SSD)
<13.3 inch, Space Grey, 1.29 kg>
Ratings - 4.7/5.0
Rs. 92,990

*Apple MacBook Pro Core i5 8th Gen (8 GB/256 GB SSD)
<13.3 inch, Space Grey, 1.37 kg>
Ratings - 4.6/5.0
Rs. 1,54,990

*Apple MacBook Pro Core i7 9th Gen (16 GB/512 GB SSD)
<16 inch, Space Grey, 2 kg>
Ratings - 4.7/5.0
Rs. 1,99,900"

* Apple MacBook Pro Corei9 8th Gen (16 GB/512 GB SSD /4 GB Graphics)
<15.4 inch, Space Grey, 1.83 kg>
Rating 4.1/5.0
Rs. 2,09,990"

*Apple MacBook Pro Core i7 9th Gen (16 GB/512 GB SSD)
<16 inch, Space Grey, 2 kg>
Ratings - 4.7/5.0
Rs. 1,99,900

*Apple MacBook Pro Core i5 10th Gen (16 GB/512 GB SSD)
<13 inch, Space Grey, 1.4 kg>
Ratings - 4.9/5.0
Rs. 1,74,900

*Apple MacBook Pro Core i5 8th Gen (8 GB/512 GB SSD)
<13.3 inch, Space Grey, 1.4 kg>
Ratings - 4.7/5.0
Rs. 1,69,900

*Apple MacBook Pro Core i9 9th Gen (16 GB/1 TB SSD/4 GB Graphics)
<16 inch, Space Grey, 2 kg>
Ratings - 4.7/5.0
Rs. 2,24,900

* Apple MacBook Pro Corei7 8th Gen (16 GB/512 GB SSD /4 GB Graphics)"<<endl;
<15.4 inch, Space Grey, 1.83 kg>
Rating 4/5.0
Rs. 2,39,900

* Apple MacBook Pro Corei5 10th Gen (16 GB/1 TB SSD )
<13 inch, Space Grey, 1.4 kg>
Rating 4.9/5.0
Rs. 1,94,900


* Thank you Visit Again :) !!!!

* Developed By ==>

* Chris(SITA-06)
* Varun(SITA-55)
* Vipul(SITA-57)
* Rutwik(SITA-61)
* Piyush(SITA-62)
```

```
D:\c_cpp_programs\projetcpp.exe
<16 inch, Space Grey, 2 kg>
Ratings - 4.7/5.0
Rs. 2,24,900

* Apple MacBook Pro Corei7 8th Gen <16 GB/512 GB SSD /4 GB Graphics>"<<endl;
<15.4 inch, Space Grey, 1.83 kg>
Rating 4/5.0
Rs. 2,39,900

* Apple MacBook Pro Corei5 10th Gen <16 GB/1 TB SSD >
<13 inch, Space Grey, 1.4 kg>
Rating 4.9/5.0
Rs. 1,94,900

* Thank you Visit Again :> !!!!
* Developed By ==>

* Chris<SITA-06>
* Varun<SITA-55>
* Vipul<SITA-57>
* Rutwik<SITA-61>
* Piyush<SITA-62>

S.Y Information Technology Department !!!!
G.H.Raisoni College of Engineering and Management,Wagholi,Pune .

User Destructed Successfully !!!!

=====

Process returned 0 (0x0)   execution time : 59.661 s
Press any key to continue.
```

**Conclusion:** Thus studied and implemented file handling for Laptop Customization System.

---