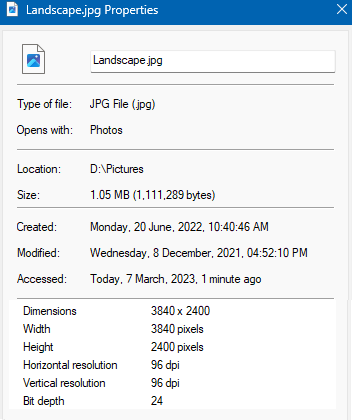
|  |
| --- |
| Object Oriented Programming |
| Course Code: CSC 210 |
| Assignment 1     |  | | --- | | Submission Instructions:   * The assignment is to be submitted in group of 2 members. * Individual submission is not allowed. For exceptions, please seek prior approval. * Submission deadline is 27 March 2023. Late submission is not allowed |  * Make sure to use this file to submit your solutions.   Submission by:  Name: Saad Ahmad Enrollment Number: 01-134222-130  Name: Saaid Jamil\_ Enrollment Number: 01-134222-134 |
|  |

## Problem Description

The objective of this assignment is to get you to carefully observe and think of a commonly used scenario and allow you to practice designing a software program using the object-oriented approach. Your task is to design and develop a software application that can store and manage information of a picture stored in a specific directory. Software can prompt users to enter information for a specific image. For each picture your application will manage information as shown in the image below.

****

Some of the fields can be changed, some may not have a value at the time of creation. Try to carefully think of each aspect to identify what information is mandatory at the time of creation of record for each picture, what information may be changed if required. There may also be some values that change automatically depending on other changes. Feel free to add data members and functions according to demand of question and your understanding.

Implement your designed Class in C++ and write down driver program in main function to show the functionality of your work. All data members must be initialized through Constructor. Destructor should print your enrollment number.

Solution should be divided into following parts.

Part a: **Entity Name and List of** **Data Members**

class Image

string name;

string type;

string filetype;

string openwith;

string location;

string created;

string modified;

string accessed;

string dimensions;

float size;

int width;

int height;

int hresolution;

int vresolution;

int bitdepth;

Part b: **List of** **Member Functions**

void setName(string);

void setType(string, string);

void setOpenwith(string);

void setLocation(string);

void setCreated(string);

void setModified(string);

void setAccessed(string);

void setDimensions(string);

void setSize(float);

void setWidth(int);

void setHeight(int);

void setHresolution(int);

void setVresolution(int);

void setBitdepth(int);

void displayData();

void modifyData();

void displayLine();

Part c: **Complete implementation of class in C++**

class Image{

//Declaration of data members

string name;

string type;

string filetype;

string openwith;

string location;

string created;

string modified;

string accessed;

string dimensions;

float size;

int width;

int height;

int hresolution;

int vresolution;

int bitdepth;

public:

~Image() //deconstructor

{

cout << "01-134222-130" << endl;

cout << "01-134222-134";

}

Image() //constructor

{

//Declaration of local variables

string a, b, c, d, f, ftype;

int i, j, k, x, y, z, choice;

//Collecting the name of the image

cout << "Enter the name of the image: ";

getline(cin >> ws, a);

setName(a);

do {

cout << "Enter 1 for JPG image, Enter 2 for PNG image :";

cin >> choice;

if (choice == 1) {

b = "JPG File(.jpg)";

ftype = ".jpg";

}

else if (choice == 2) {

b = "PNG File(.png)";

ftype = ".png";

}

else {

cout << "Invalid choice" << endl;

}

} while (choice != 1 && choice != 2);

setType(b, ftype); //Setting the type of image

c = "Photos";

setOpenwith(c);

cout << "Enter the location of the image : " << endl;

getline(cin >> ws, d);

setLocation(d); //Setting the location of the image

time\_t now = time(0); //built-in function that returns the current system time

string e;

e = ctime(&now); //built-in function that converts a time\_t (now) value into a string representation of the local time

setCreated(e);

setModified(e);

setAccessed("0");

//Crafting the size of image

cout << "Enter the width of the image : " << endl;

cin >> i;

setWidth(i);

cout << "Enter the height of the image : " << endl;

cin >> j;

setHeight(j);

cout << "Enter the horizontal resolution of the image : " << endl;

cin >> k;

setHresolution(k);

cout << "Enter the vertical resoluton of the image : " << endl;

cin >> x;

setVresolution(x);

cout << "Enter the bit depth of the image : " << endl;

cin >> y;

setBitdepth(y);

//to\_string is used to convert an integer to string

f = to\_string(i) + "x" + to\_string(j);

setDimensions(f);

float size;

//Calculates the image size

size = (i \* j \* y / 8 / 1024 / 1024);

setSize(size);

}

//Prototypes of Member functions

void setName(string);

void setType(string, string);

void setOpenwith(string);

void setLocation(string);

void setCreated(string);

void setModified(string);

void setAccessed(string);

void setDimensions(string);

void setSize(float);

void setWidth(int);

void setHeight(int);

void setHresolution(int);

void setVresolution(int);

void setBitdepth(int);

void displayData();

void modifyData();

void displayLine();

};

void Image::setName(string a) //setting name

{

name = a;

}

void Image::setType(string a, string b) //setting image type

{

type = a;

filetype = b;

}

void Image::setOpenwith(string a) //Open as photos

{

openwith = a;

}

void Image::setLocation(string a) //Sets locations

{

location = a;

}

void Image::setCreated(string a) //Exact time of creation

{

created = a;

}

void Image::setModified(string a) //Exact time of modification

{

modified = a;

}

void Image::setAccessed(string a) //The exact time accessed

{

accessed = a;

}

void Image::setDimensions(string a) //Setting dimensions

{

dimensions = a;

}

void Image::setSize(float a) //Setting size of image

{

size = a;

}

void Image::setWidth(int a) //Setting width of image

{

width = a;

}

void Image::setHeight(int a) //Setting height of image

{

height = a;

}

void Image::setHresolution(int a) //Setting horizontal resolution

{

hresolution = a;

}

void Image::setVresolution(int a) //Setting vertical resolution

{

vresolution = a;

}

void Image::setBitdepth(int a) //Setting bits

{

bitdepth = a;

}

void Image::displayLine() {

cout << "---------------------------------------------" << endl;

}

void Image::displayData()

{

//Used to get current date and time

time\_t now = time(0);

string e;

e = ctime(&now);

setModified(e);

system("CLS");

cout << "Name:" << setw(36) << name + filetype << endl; //Display name

displayLine();

cout << "Type of file:" << setw(29) << type << endl; //Display type of file

cout << "Opens with:" << setw(23) << openwith << endl;

displayLine();

cout << "Location:" << setw(30) << location << endl; //Display location

cout << "Size:" << setw(25) << size << " MB" << endl; //Display Size

displayLine();

cout << "Created:" << setw(10) << created; //Display time of creation

cout << endl;

cout << "Modified:" << setw(10) << modified; //Display time of modification

cout << endl;

time\_t now2 = time(0);

e = ctime(&now2);

setAccessed(e);

cout << "Accessed:" << setw(10) << accessed; //Display exact accessed time

displayLine();

cout << "Dimensions:" << setw(26) << dimensions << endl; //Display Dimension

cout << "Width:" << setw(26) << width << " pixels" << endl; //Display width

cout << "Height:" << setw(25) << height << " pixels" << endl; //Display Height

cout << "Horizontal resolution:" << setw(8) << hresolution << " dpi" << endl; //Display Horizontal resolution

cout << "Vertical resolution:" << setw(10) << vresolution << " dpi" << endl; //Display vertical resolution

cout << "Bit depth:" << setw(20) << bitdepth << endl; //Display bits

displayLine();

system("pause");

cout << "Enter 1 to modify data or enter any number to exit : " << endl;

int choice;

cin >> choice;

if (choice == 1)

{

modifyData(); //Modification of Data

}

else {

return;

}

}

void Image::modifyData() //Modification of Data

{

system("CLS");

cout << "Enter 1 to edit name : " << endl;

cout << "Enter 2 to edit file type : " << endl;

cout << "Enter 3 to edit location : " << endl;

cout << "Enter 4 to edit height : " << endl;

cout << "Enter 5 to edit width : " << endl;

cout << "Enter 6 to edit horizontal resolution : " << endl;

cout << "Enter 7 to edit vertical resolution : " << endl;

cout << "Enter 8 to edit bitdepth : " << endl;

int choice, c1;

cin >> choice;

string a, b, ftype, c, f;

int i, j, k, x, y;

switch (choice)

{

case 1: system("CLS");

cout << "Enter the name : ";

getline(cin >> ws, a);

setName(a); //Changing name

break;

case 2: system("CLS");

do {

cout << "Enter 1 for JPG image, Enter 2 for PNG image :";

cin >> c1;

if (c1 == 1) {

b = "JPG File(.jpg)";

ftype = ".jpg";

}

else if (c1 == 2) {

b = "PNG File(.png)";

ftype = ".png";

}

else {

cout << "Invalid choice" << endl;

}

} while (c1 != 1 && c1 != 2);

setType(b, ftype); //Chnaging file type

break;

case 3: system("CLS");

cout << "Enter the location : " << endl;

getline(cin >> ws, c);

setLocation(c); //changing location

break;

case 4: system("CLS");

cout << "Enter the height of the image : " << endl;

cin >> j;

setHeight(j); //changing height

f = to\_string(width) + "x" + to\_string(j);

setDimensions(f);

float size;

size = (width \* j \* bitdepth / 8 / 1024 / 1024);

setSize(size); //changing size

break;

case 5: system("CLS");

cout << "Enter the width of the image : " << endl;

cin >> i;

setWidth(i); //changing width

f = to\_string(i) + "x" + to\_string(height);

setDimensions(f);

size = (i \* height \* bitdepth / 8 / 1024 / 1024);

setSize(size);

break;

case 6: system("CLS");

cout << "Enter the horizontal resolution of the image : " << endl;

cin >> k;

setHresolution(k); //changing horizontal resolution

break;

case 7: system("CLS");

cout << "Enter the vertical resoluton of the image : " << endl;

cin >> x;

setVresolution(x); //changing vertical resolution

break;

case 8: system("CLS");

cout << "Enter the bit depth of the image : " << endl;

cin >> y;

setBitdepth(y); //changing bit

size = (width \* height \* y / 8 / 1024 / 1024);

setSize(size);

break;

default :

cout << "Invalid choice" << endl;

system("pause");

}

displayData();

}

Part d: **Driver Program (main function) to exhibit functionality of your class.**

int main()

{

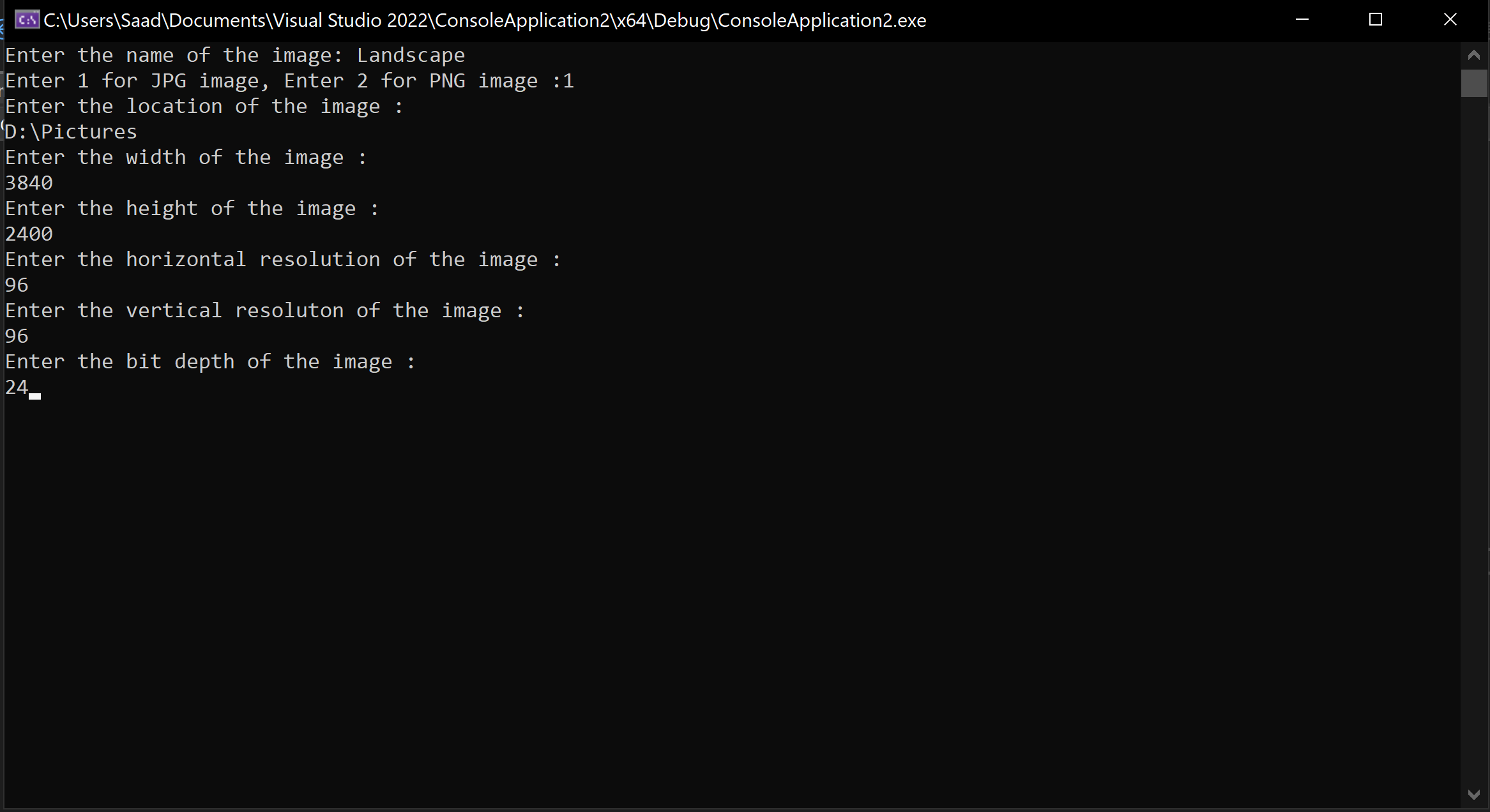
Image i1;

i1.displayData();

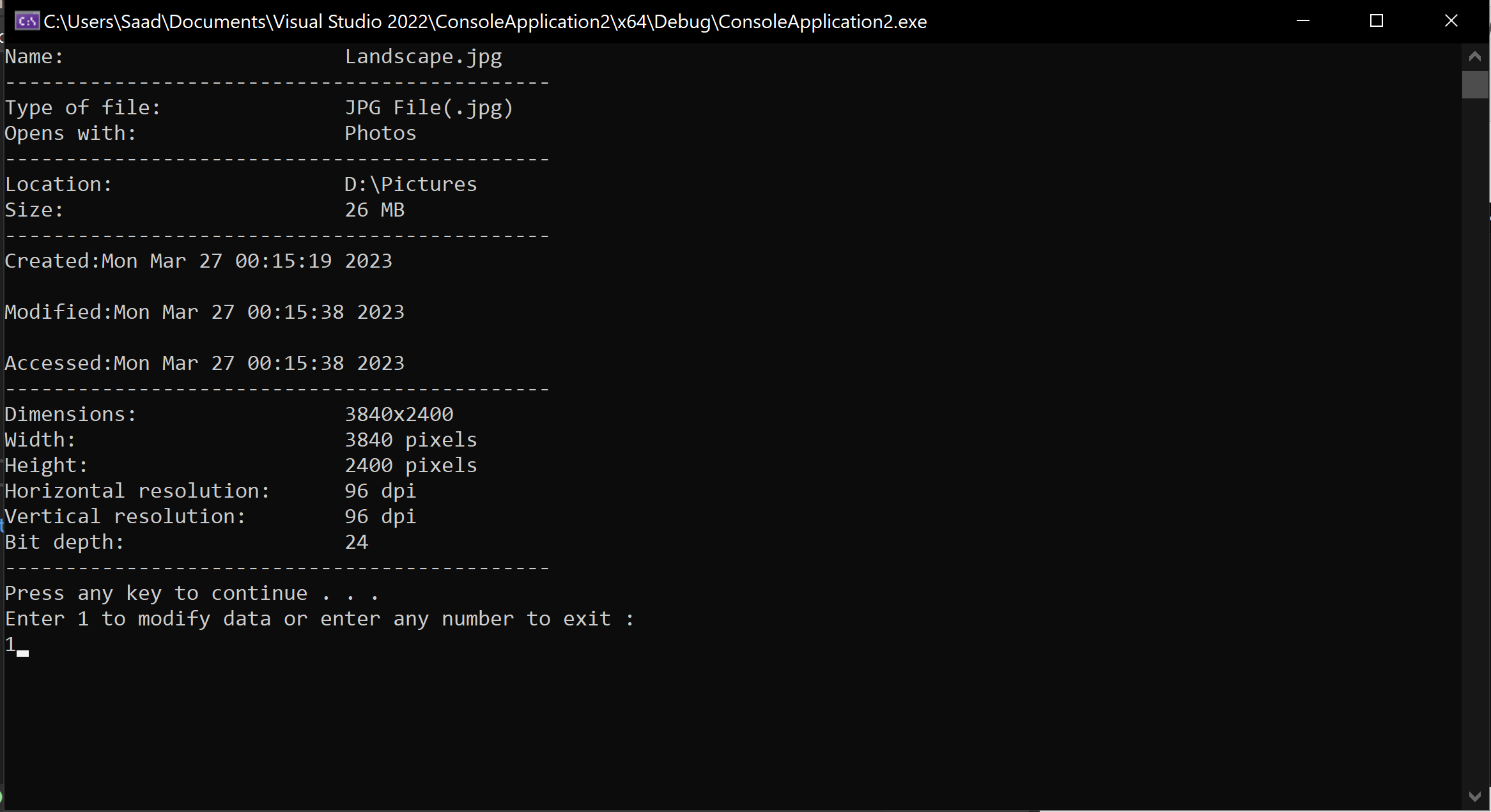
return 0;

}Part e: **Screen Shots of output**

Input Menu

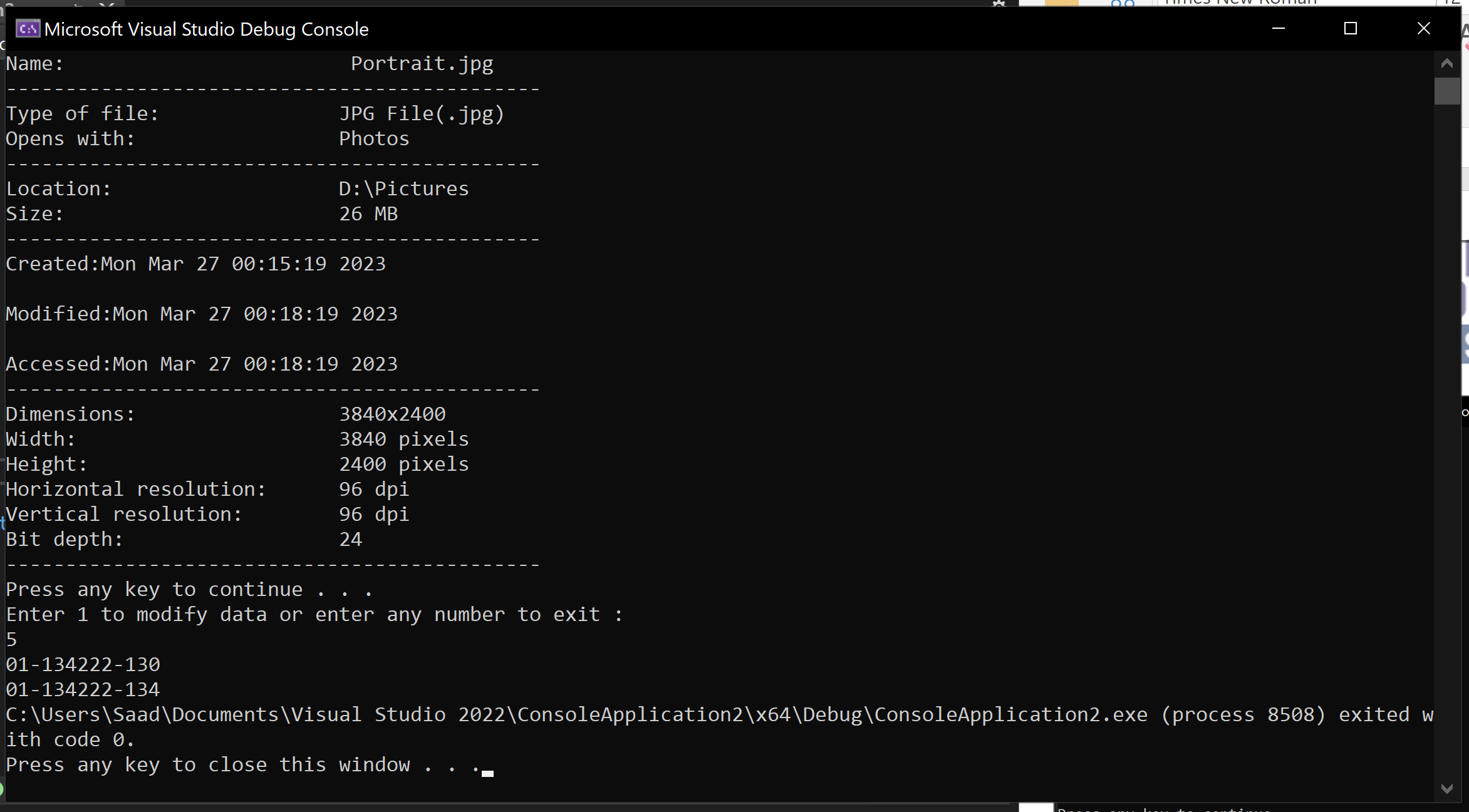


Displaying Data:



Modifying Data:





\*Remember to follow best practices, including comments, elaborative names of variables and functions.

**Code:**

#pragma warning(disable : 4996) /\* is used to provide instructions to the compiler to

disable a warning generated by the compiler when using time function.\*/

#include <iostream>

#include <string>

#include <ctime>

#include <iomanip>

using namespace std;

class Image

{

//Declaration of data members

string name;

string type;

string filetype;

string openwith;

string location;

string created;

string modified;

string accessed;

string dimensions;

float size;

int width;

int height;

int hresolution;

int vresolution;

int bitdepth;

public:

~Image() //deconstructor

{

cout << "01-134222-130" << endl;

cout << "01-134222-134";

}

Image() //constructor

{

//Declaration of local variables

string a, b, c, d, f, ftype;

int i, j, k, x, y, z, choice;

//Collecting the name of the image

cout << "Enter the name of the image: ";

getline(cin >> ws, a);

setName(a);

do {

cout << "Enter 1 for JPG image, Enter 2 for PNG image :";

cin >> choice;

if (choice == 1) {

b = "JPG File(.jpg)";

ftype = ".jpg";

}

else if (choice == 2) {

b = "PNG File(.png)";

ftype = ".png";

}

else {

cout << "Invalid choice" << endl;

}

} while (choice != 1 && choice != 2);

setType(b, ftype); //Setting the type of image

c = "Photos";

setOpenwith(c);

cout << "Enter the location of the image : " << endl;

getline(cin >> ws, d);

setLocation(d); //Setting the location of the image

time\_t now = time(0); //built-in function that returns the current system time

string e;

e = ctime(&now); //built-in function that converts a time\_t (now) value into a string representation of the local time

setCreated(e);

setModified(e);

setAccessed("0");

//Crafting the size of image

cout << "Enter the width of the image : " << endl;

cin >> i;

setWidth(i);

cout << "Enter the height of the image : " << endl;

cin >> j;

setHeight(j);

cout << "Enter the horizontal resolution of the image : " << endl;

cin >> k;

setHresolution(k);

cout << "Enter the vertical resoluton of the image : " << endl;

cin >> x;

setVresolution(x);

cout << "Enter the bit depth of the image : " << endl;

cin >> y;

setBitdepth(y);

//to\_string is used to convert an integer to string

f = to\_string(i) + "x" + to\_string(j);

setDimensions(f);

float size;

//Calculates the image size

size = (i \* j \* y / 8 / 1024 / 1024);

setSize(size);

}

//Prototypes of Member functions

void setName(string);

void setType(string, string);

void setOpenwith(string);

void setLocation(string);

void setCreated(string);

void setModified(string);

void setAccessed(string);

void setDimensions(string);

void setSize(float);

void setWidth(int);

void setHeight(int);

void setHresolution(int);

void setVresolution(int);

void setBitdepth(int);

void displayData();

void modifyData();

void displayLine();

};

void Image::setName(string a) //setting name

{

name = a;

}

void Image::setType(string a, string b) //setting image type

{

type = a;

filetype = b;

}

void Image::setOpenwith(string a) //Open as photos

{

openwith = a;

}

void Image::setLocation(string a) //Sets locations

{

location = a;

}

void Image::setCreated(string a) //Exact time of creation

{

created = a;

}

void Image::setModified(string a) //Exact time of modification

{

modified = a;

}

void Image::setAccessed(string a) //The exact time accessed

{

accessed = a;

}

void Image::setDimensions(string a) //Setting dimensions

{

dimensions = a;

}

void Image::setSize(float a) //Setting size of image

{

size = a;

}

void Image::setWidth(int a) //Setting width of image

{

width = a;

}

void Image::setHeight(int a) //Setting height of image

{

height = a;

}

void Image::setHresolution(int a) //Setting horizontal resolution

{

hresolution = a;

}

void Image::setVresolution(int a) //Setting vertical resolution

{

vresolution = a;

}

void Image::setBitdepth(int a) //Setting bits

{

bitdepth = a;

}

void Image::displayLine() {

cout << "---------------------------------------------" << endl;

}

void Image::displayData()

{

//Used to get current date and time

time\_t now = time(0);

string e;

e = ctime(&now);

setModified(e);

system("CLS");

cout << "Name:" << setw(36) << name + filetype << endl; //Display name

displayLine();

cout << "Type of file:" << setw(29) << type << endl; //Display type of file

cout << "Opens with:" << setw(23) << openwith << endl;

displayLine();

cout << "Location:" << setw(30) << location << endl; //Display location

cout << "Size:" << setw(25) << size << " MB" << endl; //Display Size

displayLine();

cout << "Created:" << setw(10) << created; //Display time of creation

cout << endl;

cout << "Modified:" << setw(10) << modified; //Display time of modification

cout << endl;

time\_t now2 = time(0);

e = ctime(&now2);

setAccessed(e);

cout << "Accessed:" << setw(10) << accessed; //Display exact accessed time

displayLine();

cout << "Dimensions:" << setw(26) << dimensions << endl; //Display Dimension

cout << "Width:" << setw(26) << width << " pixels" << endl; //Display width

cout << "Height:" << setw(25) << height << " pixels" << endl; //Display Height

cout << "Horizontal resolution:" << setw(8) << hresolution << " dpi" << endl; //Display Horizontal resolution

cout << "Vertical resolution:" << setw(10) << vresolution << " dpi" << endl; //Display vertical resolution

cout << "Bit depth:" << setw(20) << bitdepth << endl; //Display bits

displayLine();

system("pause");

cout << "Enter 1 to modify data or enter any number to exit : " << endl;

int choice;

cin >> choice;

if (choice == 1)

{

modifyData(); //Modification of Data

}

else {

return;

}

}

void Image::modifyData() //Modification of Data

{

system("CLS");

cout << "Enter 1 to edit name : " << endl;

cout << "Enter 2 to edit file type : " << endl;

cout << "Enter 3 to edit location : " << endl;

cout << "Enter 4 to edit height : " << endl;

cout << "Enter 5 to edit width : " << endl;

cout << "Enter 6 to edit horizontal resolution : " << endl;

cout << "Enter 7 to edit vertical resolution : " << endl;

cout << "Enter 8 to edit bitdepth : " << endl;

int choice, c1;

cin >> choice;

string a, b, ftype, c, f;

int i, j, k, x, y;

switch (choice)

{

case 1: system("CLS");

cout << "Enter the name : ";

getline(cin >> ws, a);

setName(a); //Changing name

break;

case 2: system("CLS");

do {

cout << "Enter 1 for JPG image, Enter 2 for PNG image :";

cin >> c1;

if (c1 == 1) {

b = "JPG File(.jpg)";

ftype = ".jpg";

}

else if (c1 == 2) {

b = "PNG File(.png)";

ftype = ".png";

}

else {

cout << "Invalid choice" << endl;

}

} while (c1 != 1 && c1 != 2);

setType(b, ftype); //Chnaging file type

break;

case 3: system("CLS");

cout << "Enter the location : " << endl;

getline(cin >> ws, c);

setLocation(c); //changing location

break;

case 4: system("CLS");

cout << "Enter the height of the image : " << endl;

cin >> j;

setHeight(j); //changing height

f = to\_string(width) + "x" + to\_string(j);

setDimensions(f);

float size;

size = (width \* j \* bitdepth / 8 / 1024 / 1024);

setSize(size); //changing size

break;

case 5: system("CLS");

cout << "Enter the width of the image : " << endl;

cin >> i;

setWidth(i); //changing width

f = to\_string(i) + "x" + to\_string(height);

setDimensions(f);

size = (i \* height \* bitdepth / 8 / 1024 / 1024);

setSize(size);

break;

case 6: system("CLS");

cout << "Enter the horizontal resolution of the image : " << endl;

cin >> k;

setHresolution(k); //changing horizontal resolution

break;

case 7: system("CLS");

cout << "Enter the vertical resoluton of the image : " << endl;

cin >> x;

setVresolution(x); //changing vertical resolution

break;

case 8: system("CLS");

cout << "Enter the bit depth of the image : " << endl;

cin >> y;

setBitdepth(y); //changing bit

size = (width \* height \* y / 8 / 1024 / 1024);

setSize(size);

break;

default :

cout << "Invalid choice" << endl;

system("pause");

}

displayData();

}

int main()

{

Image i1;

i1.displayData();

return 0;

}