C++ Practice Report

|  |  |
| --- | --- |
| Student ID | 18511160023 |
| Student Name | EMETUCHE WINNER CHIDIUTO |
| Practice No. | 6 |
| Practice Title | **Circle Classes** |
| Date |  |
| Place |  |
| Mark |  |
| Checked by |  |

# P6. Design a Circle class

## Problems

### Problem 1

Design a circle class for 2D circle.

Circle has attributes: center position(x,y), radius.

Circle has behavoirs: Get area, which can zoom, move, and two circles can be compared by its radius. greater than, equal, less than. You are required to overload comparison opeators(< > == ) for circle class based on its radius.

Test your Circle class with following program.

### Problem 2: sorting on circle class\*

Copy Input, sort, output functions from Practice P2. Dont do any changes on them. You should carefully design Circle class so that overloaded functions Input , Sort and Output applied on Circles needs a little changes, just like the other overloaded functions..

Added an overloaded functions(input, sort, output) on Circle. and use the following program to test it.

### Problem 3: use function template

Copy Input, sort, output template functions from Practice P2. Dont do any changes on them. carefully design Circle class. appy input , sort and output on this Circle class.

Apply template function (input, sort, output ) Circle. and use the following program to test it.

void Test()

{

Circle c[5];

cout<<"enter 5 circles:\n";

Input(c,5);

Sort(c,5);

cout<<" Output circles in ascending of radius\n";

Output(c,5);

}

## Results

### Result of Problem 1

**Algorithm Result**

**//----18511160023 EMETUCHE WINNER CHIDIUTO**

**//----Practice 6,**

**#include <stdlib.h>**

**#include <iostream>**

**#include <unistd.h>**

**#include <curses.h>**

**#include <string.h>**

**#include <unistd.h>**

**#include <sys/select.h>**

**#include <termios.h>**

**#include <chrono>**

**#include <ctime>**

**#include <time.h>**

**#include <iostream>**

**using namespace std;**

**class Circle{**

**double xc, yc, r;**

**public:**

**Circle(double xc1, double yc1, double r1) {**

**xc = xc1;**

**yc = yc1;**

**r = r1;**

**}**

**void Show(){**

**cout << "(" << xc << "," << yc << ") " << "," << r << endl;**

**}**

**bool operator > (Circle &c2) {**

**return this -> r > c2.r;**

**}**

**bool operator == (Circle &c2) {**

**return this -> r == c2.r;**

**}**

**void SetRadius(double r1){**

**r = r1;**

**}**

**double GetRadius() {**

**return r;**

**}**

**};**

**int main(){**

**double x1, y1, r1;**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* Testing Instructions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* Please Enter test Values for Two Circles \*\*\*\*\*\*"<<endl;**

**cout << "Enter First circle, center and radius in Format [ circle center radius ]\n";**

**cout << "Example: 2.3 3 5 \n";**

**cin >> x1 >> y1 >> r1;**

**Circle c1(x1, y1, r1); // 3 and 5 is center , 2.5 is radius;**

**cout << "Enter Second circle, center and radius in Format [ circle center radius ]\n";**

**cin >> x1 >> y1 >> r1;**

**Circle c2(x1, y1, r1); // 3 and 5 is center , 2.5 is radius;**

**cout << "c1=";**

**c1.Show();**

**cout << "c2=";**

**c2.Show();**

**if (c1> c2)**

**cout << " c1 is bigger than c2" << endl;**

**else if (c1 == c2)**

**cout << " c1 is equal to c2" << endl;**

**else**

**cout << " c1 is smaller than c2" << endl;**

**r1 = rand() % 50;**

**c1.SetRadius(r1);**

**r1 = rand() % 50;**

**c2.SetRadius(r1);**

**cout << "c1=";**

**c1.Show();**

**cout << "c2=";**

**c2.Show();**

**if (c1> c2){**

**cout << " c1 is Bigger than c2" << endl;**

**}else if (c1 == c2){**

**cout << " c1 is Equal to c2" << endl;**

**}else{**

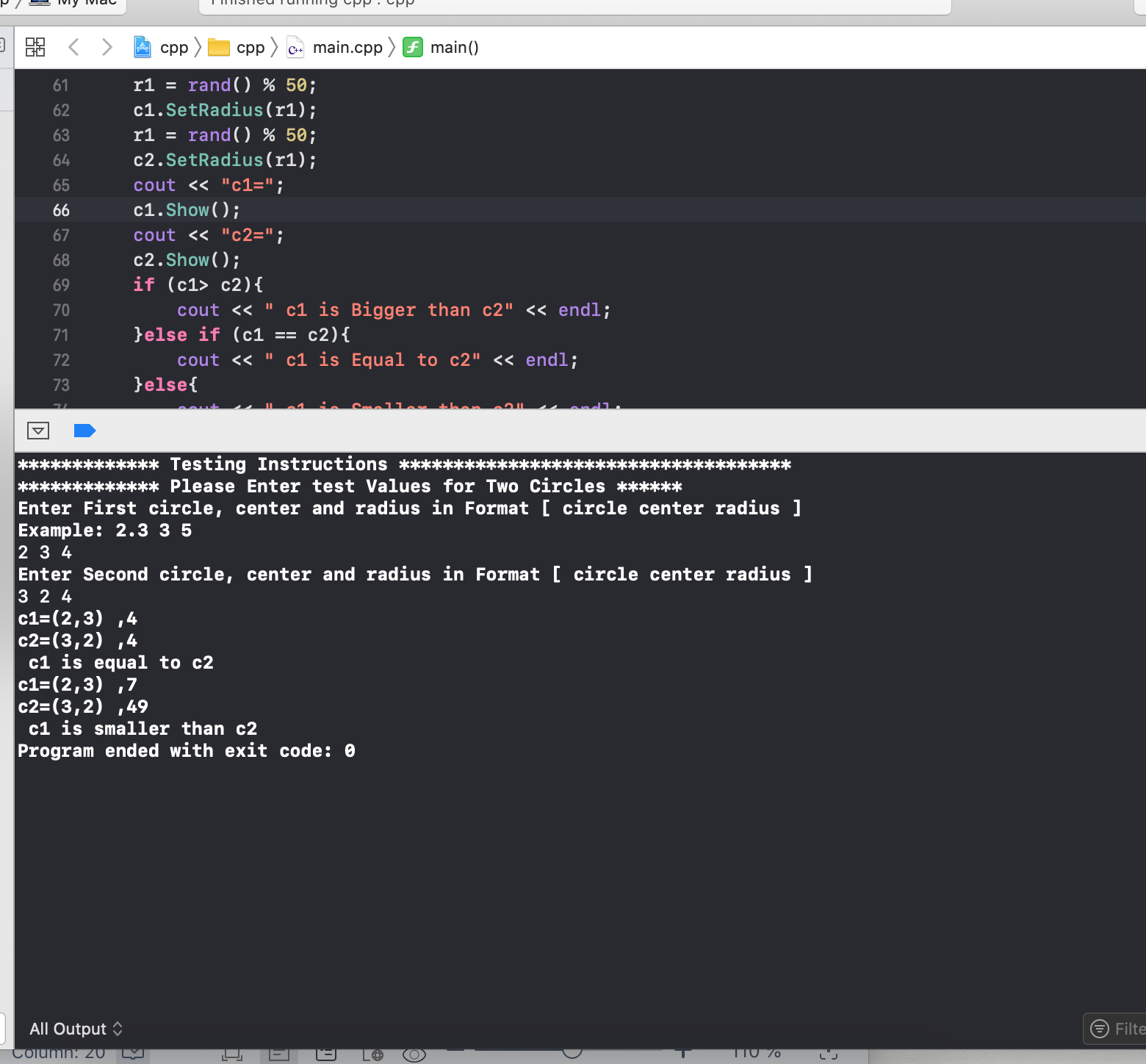
**cout << " c1 is Smaller than c2" << endl;**

**}**

**return 0;**

**}**

[Screen shot]



### Result of Problem 2

**Algorithm Result**

**//----18511160023 EMETUCHE WINNER CHIDIUTO**

**//----Practice 6,**

**#include <stdlib.h>**

**#include <iostream>**

**#include <unistd.h>**

**#include <curses.h>**

**#include <string.h>**

**#include <unistd.h>**

**#include <sys/select.h>**

**#include <termios.h>**

**#include <chrono>**

**#include <ctime>**

**#include <time.h>**

**#include <iostream>**

**using namespace std;**

**class Circle{**

**double xc, yc, r;**

**public:**

**Circle (){}**

**Circle(double xc1, double yc1, double r1) {}**

**void Show(){}**

**int operator > (Circle &c2) {**

**return this->r > c2.r;**

**}**

**int operator ==(Circle &c2) {**

**return this->r == c2.r;**

**}**

**void SetRadius(double r1) {}**

**double GetRadius() {**

**return r;**

**}**

**friend istream &operator>>(istream &in, Circle &c){**

**in >> c.xc >> c.yc >> c.r ;**

**return in;**

**}**

**friend ostream &operator<<(ostream &on, Circle &c){**

**on << "(" << c.xc << "," << c.yc << ")" << "," << c.r << endl;**

**return on;**

**}**

**};**

**void Input(Circle a[], int n){**

**int i;**

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

**cin >> a[i];**

**}**

**}**

**void Output(Circle a[], int n){**

**int i;**

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

**cout << a[i] << endl;**

**}**

**}**

**void Sort(Circle c[], int n){**

**Circle swapper;**

**int outterSequence, innerSequence;**

**for(outterSequence=0;outterSequence<n-1;outterSequence++){**

**for(innerSequence=0; innerSequence<n-outterSequence-1; innerSequence++){**

**if (c[innerSequence] > c[innerSequence + 1]){**

**swapper = c[innerSequence];**

**c[innerSequence] = c[innerSequence + 1];**

**c[innerSequence + 1] = swapper;**

**}**

**}**

**}**

**}**

**int main(){**

**Circle c[5];**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* Testing Instructions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* You have to Enter Values for 5 Circles \*\*\*\*\*\*"<<endl;**

**cout << "enter 5 circles:\n";**

**Input(c, 5);**

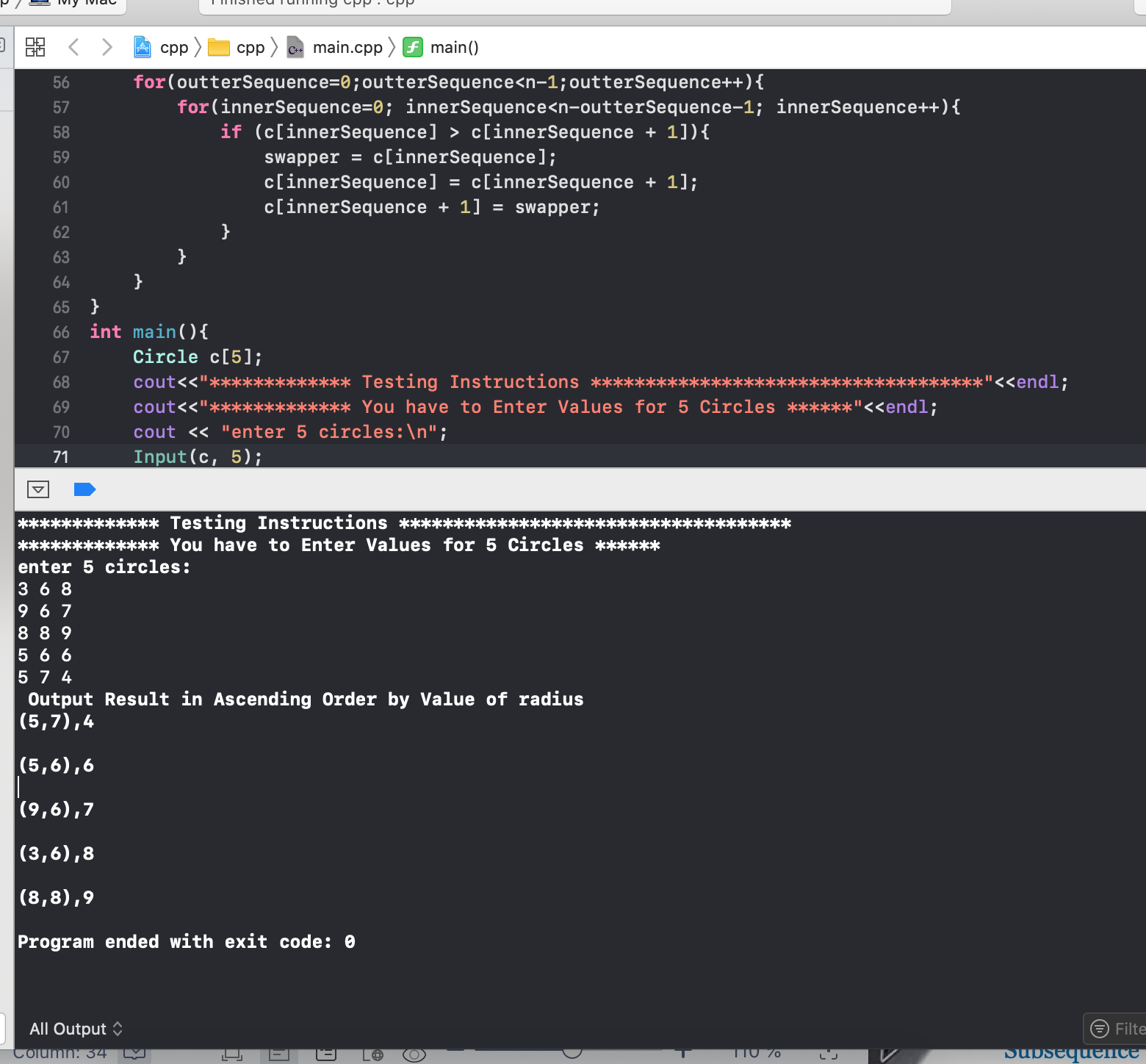
**Sort(c, 5);**

**cout << " Output Result in Ascending Order by Value of radius\n";**

**Output(c, 5);**

**}**

**Screen shot Result:**



### Result of Problem 2

**Algorithm Result:**

**//----18511160023 EMETUCHE WINNER CHIDIUTO**

**//----Practice 6,**

**#include <stdlib.h>**

**#include <iostream>**

**#include <unistd.h>**

**#include <curses.h>**

**#include <string.h>**

**#include <unistd.h>**

**#include <sys/select.h>**

**#include <termios.h>**

**#include <chrono>**

**#include <ctime>**

**#include <time.h>**

**#include <iostream>**

**using namespace std;**

**class Circle{**

**double xc, yc, r;**

**public:**

**Circle (){}**

**Circle(double xc1, double yc1, double r1){}**

**void Show(){}**

**int operator > (Circle &c2) {**

**return this->r > c2.r;**

**}**

**int operator ==(Circle &c2) {**

**return this->r == c2.r;**

**}**

**void SetRadius(double r1) {}**

**double GetRadius() {**

**return r;**

**}**

**friend istream &operator>>(istream &in, Circle &c){**

**in >> c.xc >> c.yc >> c.r ;**

**return in;**

**}**

**friend ostream &operator<<(ostream &on, Circle &c){**

**on << "(" << c.xc << "," << c.yc << ")" << "," << c.r << endl;**

**return on;**

**}**

**};**

**template<class mabasa>**

**void Input(mabasa a[], int n){**

**int i;**

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

**cin >> a[i];**

**}**

**template<class mabasa>**

**void Output(mabasa a[], int n){**

**int i;**

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

**cout << a[i] << endl;**

**}**

**}**

**template<class mabasa>**

**void Sort(mabasa c[], int n){**

**Circle swapper;**

**int outterSequence, innerSequence;**

**for(outterSequence=0;outterSequence<n-1;outterSequence++){**

**for(innerSequence=0;innerSequence<n-outterSequence-1;innerSequence++){**

**if (c[innerSequence] > c[innerSequence + 1]){**

**swapper = c[innerSequence];**

**c[innerSequence] = c[innerSequence + 1];**

**c[innerSequence + 1] = swapper;**

**}**

**}**

**}**

**}**

**int main(){**

**Circle c[5];**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* Testing Instructions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;**

**cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* You have to Enter Values for 5 Circles \*\*\*\*\*\*"<<endl;**

**Input(c, 5);**

**Sort(c, 5);**

**cout << " Output Result in Ascending Order by Value of radius\n";**

**Output(c, 5);**

**}**

**Screen shot resut**

