Experiment 8

Exercise 2:

(15 points)

Every Circle has a center and a radius. Create a class CircleType that can store the center, the radius, and the color of the circle. Since the center of a circle is a point in the x-y plane, create a class PointType to store the x and y coordinate. Use class PointType to define the class CircleType. Provide constructors that enable objects of these classes to be initialized when they are declared. The constructors should contain default values in case no initializes are provided. The definition of class CircleType and class PointType is as under: (you may define additional functions if you require any).

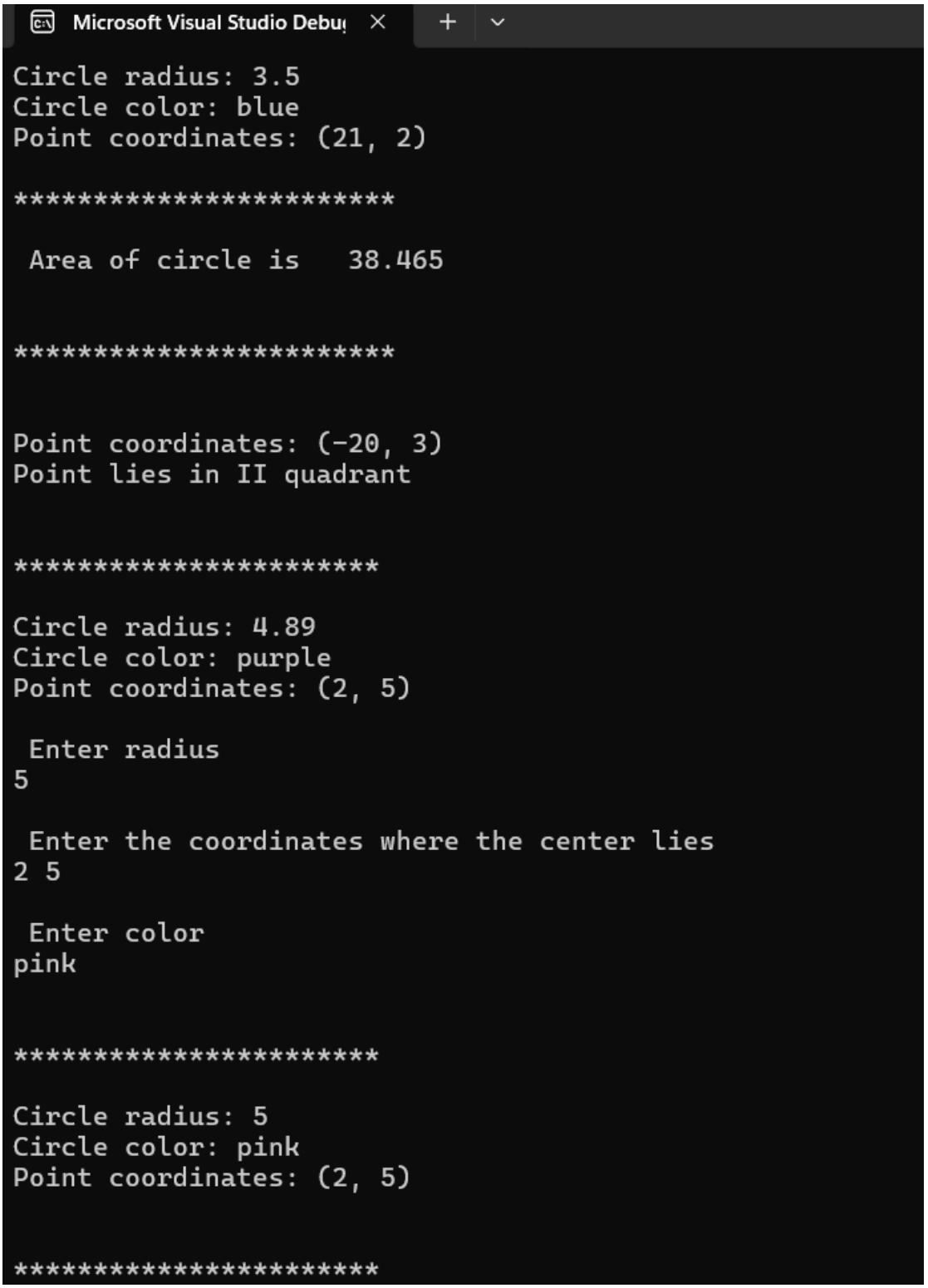
```
#include <iostream>
#include <cstring>
#include <cmath>
using namespace std;
class PointType {
    int x;
    int y;

public:

    PointType() : x(0), y(0) {}


    PointType(int x_val, int y_val) : x(x_val), y(y_val) {}


    int getX() const { return x; }
    int getY() const { return y; }


    void setX(int x_val) { x = x_val; }
    void setY(int y_val) { y = y_val; }


    void print() const {
        cout << "Point coordinates: (" << x << ", " << y << ")" << endl;
    }


    int checkquad() const {
        if (x == 0 && y == 0)
```

```
            return 0;
        else if (x > 0 && y > 0)
            return 1;
        else if (x < 0 && y > 0)
            return 2;
        else if (x < 0 && y < 0)
            return 3;
        else if (x > 0 && y < 0)
            return 4;
        else
            return -1;
    }
};

class CircleType {
    double radius;
    char* color;
    PointType center;

public:

    CircleType() : radius(0), color(nullptr) {}


    CircleType(int x, int y, double r, const char* c) : radius(r),
color(nullptr), center(x, y) {

        int len = strlen(c);
        color = new char[len + 1];
        for (int i = 0; i < len; ++i) {
            color[i] = c[i];    //copies the input string c into the color member

        }
        color[len] = '\0';  //null terminates the string
    }

    // Destructor automatically when object out of scope
    ~CircleType() {
        delete[] color;
    }

    //print function
    void print() const {
        std::cout << "Circle radius: " << radius << std::endl;
        std::cout << "Circle color: " << color << std::endl;
```

```
        center.print();
    }


    double calc_area() const {
        return 3.14 * radius * radius;
    }


    double calc_circumference() const {
        return 2 * 3.14 * radius;
    }


    void setparam(int x, int y, double r, const char* c) {
        center.setX(x);
        center.setY(y);
        radius = r;
        delete[] color;

        int len = strlen(c);
        color = new char[len + 1];
        for (int i = 0; i < len; ++i) {
            color[i] = c[i];
        }
        color[len] = '\0';
    }
};

int main() {
    CircleType C(21, 2, 3.5, "blue");
    cout << "\n**********************\n" << endl;
    C.print();
    cout << "\n**********************\n" << endl;
    cout << " Area of circle is " << C.calc_area() << endl;
    cout << "\n\n**********************\n\n" << endl;
    PointType P(-20, 3);
    int p = P.checkquad();
    P.print();
    switch (p) {
    case 0:
        cout << "Point lies at center" << endl;
        break;
    case 1:
        cout << "Point lies in I quadrant" << endl;
```

```
        break;
    case 2:
        cout << "Point lies in II quadrant" << endl;
        break;
    case 3:
        cout << "Point lies in III quadrant" << endl;
        break;
    case 4:
        cout << "Point lies in IV quadrant" << endl;
        break;
    default:
        cout << "INVALID";
        break;
    }

    double r;
    int x, y;
    char col[9];
    CircleType circ(2, 5, 4.89, "purple");
    cout << "\n\n**********************\n\n";
    circ.print();
    cout << "\n Enter radius \n";
    cin >> r;
    cout << "\n Enter the coordinates where the center lies \n";
    cin >> x >> y;
    cout << "\n Enter color \n";
    cin >> col;
    circ.setparam(x, y, r, col);
    cout << "\n\n**********************\n\n";
    circ.print();
    cout << "\n\n**********************\n\n";
    return 0;

}
```

Output:


Microsoft Visual Studio Debu
Circle radius: 3.5
Circle color: blue
Point coordinates: (21, 2)
****
Area of circle is 38.465
****
Point coordinates: (-20, 3)
Point lies in II quadrant
****
Circle radius: 4.89
Circle color: purple
Point coordinates: (2, 5)
Enter radius
5
Enter the coordinates where the center lies
2 5
Enter color
pink
****
Circle radius: 5
Circle color: pink
Point coordinates: (2, 5)
****