OOPS

Class and Object

#include <bits/stdc++.h>

using namespace std;

class rectangle

{

    int length;

    int breadth;

    int area()

    {

        return length \* breadth;

    }

    int perimeter()

    {

        return 2 \* (length + breadth);

    }

};

int main()

{

    rectangle a, b;

    a.length = 10;

    a.breadth = 15;

    cout << a.area() << endl;

    cout<<a.perimeter()<<endl;

    return 0;

}

Pointer to Object

int main()

{

    rectangle r;

    rectangle \*p;

    p = &r;

    p->length = 10;

    p->breadth = 15;

    cout << p->area() << endl;

    // Creating an object in Heap

    rectangle \*q = new rectangle();

    q->length = 10;

    q->breadth = 15;

    cout << q->area() << endl;

    return 0;

}

Access Modifiers:

i) Public: All the class members with public modifier can be accessed from anywhere (inside and outside the class).

ii) Private: All the class members with private modifier can only be accessed by the member function inside the class.

iii) Protected: The access level of a protected modifier is within the class and outside the class through child class. If you do not make the child class, it cannot be accessed from outside the class.

Get and Set

class rectangle

{

    int length;

    int breadth;

public:

    void setlength(int l)

    {

        if (l >= 0)

        {

            length = l;

        }

        else

        {

            l = 0;

        }

    }

    void setbreadth(int b)

    {

        if (b >= 0)

        {

            breadth = b;

        }

        else

        {

            breadth = 0;

        }

    }

    int getlength()

    {

        return length;

    }

    int setbreadth()

    {

        return breadth;

    }

    int area()

    {

        return length \* breadth;

    }

    int perimeter()

    {

        return 2 \* (length + breadth);

    }

};

int main()

{

    rectangle r;

    r.setlength(10);

    r.setbreadth(15);

    cout << r.area() << endl;

    return 0;

}