```cpp
//using objects in code

#include <iostream>
#include <iomanip>
using namespace std;
#define PI 3.14159
class Circle {
private:
    double radius;
public:
    void setRad(void);
    double calPer(void);
    double calAre(void);
};
void prnAreandPer(Circle x); //takes  object as argument
Circle defCirc(void); // returns an object

int main(void) {

    Circle c1, c2, c3;

    cout << setiosflags(ios::fixed | ios::showpoint);
    cout << setprecision(1);

    c1.setRad();

    cout << "Perimeter is: " << c1.calPer() << endl;
    cout << "Area is : " << c1.calAre() << endl;


    c2 = c1; // assigning an object to another

    cout << "Perimeter is: " << c2.calPer() << endl;
    cout << "Area is : " << c2.calAre() << endl;


    prnAreandPer(c2);

    c3 = defCirc();

    prnAreandPer(c3);

    return 0;
}
void Circle::setRad(void) {
    cout << "Please enter radius: ";
    cin >> radius;
}
double Circle::calPer(void) {
    return 2 * PI * radius;
}
double Circle::calAre(void) {
    return PI * radius * radius;
```

```cpp
}

void prnAreandPer(Circle x) {  //assigning values of c1 to c2
    cout << "Perimeter is: " << x.calPer() << endl;
    cout << "Area is : " << x.calAre() << endl;



}
Circle defCirc(void) {
    Circle x;
    x.setRad();
    return(x);
}


//access functions
#include <iostream>
#include <iomanip>
using namespace std;
class Rect {
private:
    double a;
    double b;
public:
    void setA(void); //access function
    void setB(void); //access function
    double getA(void); //access function
    double getB(void); //access function
    double calcArea(void);
    double calcPer(void);
};
int main(void) {
    Rect r1;

    r1.setA();
    r1.setB();

    cout << setiosflags(ios::fixed | ios::showpoint);
    cout << setprecision(1);

    cout << "For a rectangle with sides ";

    cout << r1.getA() << " and " << r1.getB() << endl;

    cout << "\tArea = " << r1.calcArea() << endl;

    cout << "\tPerimeter = " << r1.calcPer() << endl;

    return 0;
}
void Rect::setA(void) {
    cout << "Please enter side A: ";
    cin >> a;
}
void Rect::setB(void) {
    cout << "Please enter side B: ";
    cin >> b;
}
double Rect::getA(void) {
    return a;
}
double Rect::Rect::getB(void) {
    return b;
}
double Rect::calcArea(void) {
    return a * b;
}
double Rect::calcPer(void) {
    return 2 * (a + b);
}


//utility helper functions
#include <iostream>
#include <iomanip>

using namespace std;
class Time {
private:
    int hour;
    int minute;
    void convTo12(void); // Helper function //availble only to class members
                         //u cant access it in main
public:
    Time(int h = 23, int m = 59); //default values of constructor
    void prnTime(void);
};

int main(void) {
    Time t1, t2(1, 1),t3(24); //t1 has default values of 23 59 and t2 has 1 1
    t1.prnTime(); // 11:59
    t2.prnTime(); // 01:01
    t3.prnTime(); // 00:59
    return 0;
}
Time::Time(int h, int m) {   //constructor
    hour = h;
    minute = m;
}
void Time::prnTime(void) {
    convTo12();
    cout << setfill('0');
    cout << setw(2) << hour << ":";
    cout << setw(2) << minute << endl;
```

```cpp
}
void Time::convTo12(void) {
    hour = hour % 12;  //23 % 12 = 11
}


/**********************/

#include<iostream>
using namespace std;
class Complex {
private:
    int real;
    int imag;
public:
    void setvalue();
    void display();
    void sum(Complex c1, Complex c2); //PASSING TWO OBJECTS

};
int main()
{
    Complex c1, c2, c3;

    cout << "Enter real and imaginary part of first complex number"  << endl;
    c1.setvalue();

    cout << "Enter real and imaginary part of second complex number"  << endl;
    c2.setvalue();

    cout << "Sum of two complex numbers is"  << endl;
    c3.sum(c1, c2);

    c3.display();

    return 0;
}
/* Function to set the values of
    * real and imaginary part of each complex number
    */
void Complex::setvalue()
{
    cin >> real;
    cin >> imag;
}
/* Function to display the sum of two complex numbers */
void Complex::display()
{
    cout << real << "+" << imag << "i" << endl;
}
/* Function to add two complex numbers */

```

```cpp
void Complex::sum(Complex c1, Complex c2)
{
    real = c1.real + c2.real;
    imag = c1.imag + c2.imag;
}

```