

```
1 //using objects in code
2
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6 #define PI 3.14159
7 class Circle {
8 private:
9     double radius;
10 public:
11     void setRad(void);
12     double calPer(void);
13     double calAre(void);
14 };
15 void prnAreandPer(Circle x); //takes object as argument
16 Circle defCirc(void); // returns an object
17
18 int main(void) {
19
20     Circle c1, c2, c3;
21
22     cout << setiosflags(ios::fixed | ios::showpoint);
23     cout << setprecision(1);
24
25     c1.setRad();
26
27     cout << "Perimeter is: " << c1.calPer() << endl;
28     cout << "Area is : " << c1.calAre() << endl;
29
30
31     c2 = c1; // assigning an object to another
32
33     cout << "Perimeter is: " << c2.calPer() << endl;
34     cout << "Area is : " << c2.calAre() << endl;
35
36
37     prnAreandPer(c2);
38
39     c3 = defCirc();
40
41     prnAreandPer(c3);
42
43     return 0;
44 }
45 void Circle::setRad(void) {
46     cout << "Please enter radius: ";
47     cin >> radius;
48 }
49 double Circle::calPer(void) {
50     return 2 * PI * radius;
51 }
52 double Circle::calAre(void) {
53     return PI * radius * radius;
```

```
54 }
55
56 void prnAreandPer(Circle x) { //assigning values of c1 to c2
57     cout << "Perimeter is: " << x.calPer() << endl;
58     cout << "Area is : " << x.calAre() << endl;
59
60
61
62 }
63 Circle defCirc(void) {
64     Circle x;
65     x.setRad();
66     return(x);
67 }
68
69
70 //access functions
71 #include <iostream>
72 #include <iomanip>
73 using namespace std;
74 class Rect {
75 private:
76     double a;
77     double b;
78 public:
79     void setA(void); //access function
80     void setB(void); //access function
81     double getA(void); //access function
82     double getB(void); //access function
83     double calcArea(void);
84     double calcPer(void);
85 };
86 int main(void) {
87     Rect r1;
88
89     r1.setA();
90     r1.setB();
91
92     cout << setiosflags(ios::fixed | ios::showpoint);
93     cout << setprecision(1);
94
95     cout << "For a rectangle with sides ";
96
97     cout << r1.getA() << " and " << r1.getB() << endl;
98
99     cout << "\tArea = " << r1.calcArea() << endl;
100
101     cout << "\tPerimeter = " << r1.calcPer() << endl;
102
103     return 0;
104 }
105 void Rect::setA(void) {
106     cout << "Please enter side A: ";
```

```
107     cin >> a;
108 }
109 void Rect::setB(void) {
110     cout << "Please enter side B: ";
111     cin >> b;
112 }
113 double Rect::getA(void) {
114     return a;
115 }
116 double Rect::Rect::getB(void) {
117     return b;
118 }
119 double Rect::calcArea(void) {
120     return a * b;
121 }
122 double Rect::calcPer(void) {
123     return 2 * (a + b);
124 }
125
126
127 //utility helper functions
128 #include <iostream>
129 #include <iomanip>
130
131 using namespace std;
132 class Time {
133 private:
134     int hour;
135     int minute;
136     void convTo12(void); // Helper function //available only to class members
137                         //u cant access it in main
138 public:
139     Time(int h = 23, int m = 59); //default values of constructor
140     void prnTime(void);
141 };
142
143 int main(void) {
144     Time t1, t2(1, 1), t3(24); //t1 has default values of 23 59 and t2 has 1 1
145     t1.prnTime(); // 11:59
146     t2.prnTime(); // 01:01
147     t3.prnTime(); // 00:59
148     return 0;
149 }
150 Time::Time(int h, int m) { //constructor
151     hour = h;
152     minute = m;
153 }
154 void Time::prnTime(void) {
155     convTo12();
156     cout << setfill('0');
157     cout << setw(2) << hour << ":";
158     cout << setw(2) << minute << endl;
```

```
159 }
160 void Time::convTo12(void) {
161     hour = hour % 12; //23 % 12 = 11
162 }
163
164
165 /*****
166
167 #include<iostream>
168 using namespace std;
169 class Complex {
170 private:
171     int real;
172     int imag;
173 public:
174     void setvalue();
175     void display();
176     void sum(Complex c1, Complex c2); //PASSING TWO OBJECTS
177
178 };
179 int main()
180 {
181     Complex c1, c2, c3;
182
183     cout << "Enter real and imaginary part of first complex number" << endl;
184     c1.setvalue();
185
186     cout << "Enter real and imaginary part of second complex number" <<      ↗
187         endl;
188     c2.setvalue();
189
190     cout << "Sum of two complex numbers is" << endl;
191     c3.sum(c1, c2);
192     c3.display();
193
194     return 0;
195 }
196 /* Function to set the values of
197    * real and imaginary part of each complex number
198    */
199 void Complex::setvalue()
200 {
201     cin >> real;
202     cin >> imag;
203 }
204 /* Function to display the sum of two complex numbers */
205 void Complex::display()
206 {
207     cout << real << "+" << imag << "i" << endl;
208 }
209 /* Function to add two complex numbers */
210
```

---

```
211 void Complex::sum(Complex c1, Complex c2)
212 {
213     real = c1.real + c2.real;
214     imag = c1.imag + c2.imag;
215 }
216
```