```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;
}
```

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

```
107     cin >> b;
108 }
109 double Rect::getA(void) {
110     return a;
111 }
112 double Rect::Rect::getB(void) {
113     return b;
114 }
115 double Rect::calcArea(void) {
116     return a * b;
117 }
118 double Rect::calcPer(void) {
119     return 2 * (a + b);
120 }
121
122
123 //utility helper functions
124 #include <iostream>
125 #include <iomanip>
126
127 using namespace std;
128 class Time {
129 private:
130     int hour;
131     int minute;
132     void convTo12(void); // Helper function //availble only to class members
133                          //u cant access it in main
134 public:
135     Time(int h = 23, int m = 59); //default values of constructor
136     void prnTime(void);
137 };
138
139 int main(void) {
140     Time t1, t2(1, 1),t3(24); //t1 has default values of 23 59 and t2 has 1
        1
141     t1.prnTime(); // 11:59
142     t2.prnTime(); // 01:01
143     t3.prnTime(); // 00:59
144     return 0;
145 }
146 Time::Time(int h, int m) {   //constructor
147     hour = h;
148     minute = m;
149 }
150 void Time::prnTime(void) {
151     convTo12();
152     cout << setfill('0');
153     cout << setw(2) << hour << ":";
154     cout << setw(2) << minute << endl;
155 }
156 void Time::convTo12(void) {
157     hour = hour % 12;  //23 % 12 = 11
158 }
```

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

```
211 }
212
```