

```

1 // Example C++ Classes
2 // George F. Riley, Georgia Tech, Spring 2009
3
4 #include <iostream> // Needed for the "cout" commands
5
6 using namespace std;
7
8 // Define a C Structure called TwoInt_t
9 typedef struct
10 {
11     int a;
12     int b;
13 } TwoInt_t;
14
15 void PrintTwo(TwoInt_t* ti)
16 {
17     cout << "a " << ti->a << " b " << ti->b << endl;
18 }
19
20 // Define a second structure with TwoInt_t as an element
21 typedef struct
22 {
23     TwoInt_t ti;
24     int      c;
25     int      d;
26 } FourInt_t;
27
28 void PrintFour(FourInt_t* fi)
29 {
30     cout << "a " << fi->ti.a << " b " << fi->ti.b
31         << " c " << fi->c << " d " << fi->d << endl;
32 }
33
34 // Define a C++ Class that does the same thing as TwoInt_t
35 class TwoInt
36 {
37 public:
38     int a;
39     int b;
40 };
41
42 void PrintTwo(TwoInt* ti)
43 {
44     cout << "a " << ti->a << " b " << ti->b << endl;
45 }
46
47 // Define a C++ Class that does the same thing as TwoInt_t
48 // plus a "Constructor"
49 class TwoIntC
50 {
51 public:
52     TwoIntC();
53 public:
54     int a;
55     int b;
56 };

```

Program classes.cc

```

57
58
59 // Code for TwoIntC constructor
60 TwoIntC::TwoIntC()
61 {
62     a = 1000;
63     b = 1001;
64 }
65
66 void PrintTwo(TwoIntC* ti)
67 {
68     cout << "a " << ti->a << " b " << ti->b << endl;
69 }
70
71 // Demonstrate a C++ "reference" argument.
72 // More about this later in the semester
73 void PrintTwo(const TwoIntC& ti)
74 {
75     cout << "a " << ti.a << " b " << ti.b << endl;
76 }
77
78 // Make a subclass of TwoIntC called FourInt
79 class FourInt : public TwoIntC {
80 public:
81     FourInt(); // Constructor
82 public:
83     int c;
84     int d;
85 };
86
87
88 // Code for FourInt constructor
89 FourInt::FourInt()
90     : TwoIntC()
91 {
92     c = 3001;
93     d = 3002;
94 }
95
96 // Define a subroutine to print FourInt variables c and d
97 void PrintFour(FourInt* fi)
98 {
99     cout << "c " << fi->c << " d " << fi->d << endl;
100 }
101
102
103 int main()
104 {
105     TwoInt_t a; // Uninitialized
106     // Print the uninitialized one
107     PrintTwo(&a);
108
109     // Declare another one, and initialize it
110     TwoInt_t b; // Initialized below
111     b.a = 1;
112     b.b = 2;

```

Program classes.cc (continued)

```

113     PrintTwo(&b);
114
115     // Declare a FourInt variable
116     FourInt_t f;
117     f.ti.a = 10;
118     f.ti.b = 11;
119     f.c     = 12;
120     f.d     = 13;
121     PrintFour(&f);
122
123     // Now try a trick. Pass the FourInt variable to PrintTwo
124     PrintTwo((TwoInt_t*)&f);
125     //PrintTwo(&f);
126
127     // Declare a variable of class TwoInt
128     TwoInt cTi;
129     // Since we have no constructor in class TwoInt, a and b are random
130     PrintTwo(&cTi);
131     // Initialize a and b to good values.
132     cTi.a = 20;
133     cTi.b = 21;
134     PrintTwo((TwoInt_t*)&cTi);
135
136     // Declare a variable of class TwoIntC
137     TwoIntC cTiC;
138     PrintTwo(&cTiC);
139     // Call the PrintTwo with the reference argument
140     PrintTwo(cTiC);
141
142     // Declare a variable of class FourInt
143     FourInt fi;
144     PrintTwo(&fi); // Note we are calling the "TwoInt" printer
145     PrintFour(&fi);
146 }

```

Program classes.cc (continued)