

```

1 // Demonstrate more various high-level syntax for the C language
2 // ECE2036
3 // George F. Riley, Georgia Tech, Fall 2012
4
5 // Include the C++ output library "iostream"
6
7 #include <iostream>
8
9 // Illustrate the definition of a C/C++ "enumeration type"
10 // This simply defines several symbols that can be referred to by
11 // name in the program, and the symbols are given unique integer
12 // identifiers. Not this DOES NOT create a variable, but rather
13 // defines a type. We will use this type later in the "switch"
14 // example.
15
16 typedef enum
17 {
18     Yugo,
19     Ford,
20     Chevrolet,
21     Chrysler,
22     Toyota,
23     Honda,
24     Mercedes,
25     Ferrari
26 } CarModels_t;
27
28 // Define the MyRec structure (from handout 1)
29 typedef struct
30 { // myStruct has three subvariables (or components).
31     // Note the components can be different types, but don't have to be
32     int    a;
33     char   b;
34     double c;
35 } myStruct;
36
37
38 // Implement a PriceModel function that uses a switch statement
39 // to select one of several alternatives based on the enumeration
40 // type argument passed in as an argument, and return the approximate
41 // cost of the model
42 double PriceModel(CarModels_t model)
43 {
44     // The switch statement selects one of a list of alternatives
45     double cost;
46     switch (model) {
47         case Yugo:
48             cost = 2000.00; // Cost for the Yugo
49             break;         // break is needed, otherwise "falls through" to Ford
50         case Ford:
51             cost = 18000.00; // Cost for the Ford
52             break;
53         case Chevrolet:
54             cost = 19000.00;
55             break;
56         case Chrysler:

```

Program csyntax1.cc

```

57     cost = 19500.00;
58     break;
59     case Toyota:
60         cost = 20000.00;
61         break;
62     case Honda:
63         cost = 20000.00;
64         break;
65     case Mercedes:
66         cost = 50000.00;
67         break;
68     case Ferrari:
69         cost = 150000.00;
70         break;
71     }
72     return cost;
73 }
74
75
76
77 // Implement a "void" function called "Print1". The "void" return type
78 // simply says the function does not return any value.
79 // In this example, we pass a pointer to a character array
80 // that is terminated with the '\0' character.
81 // This also demonstrates the use of the de-referencing operator
82 // and the pointer increment (++)
83 //
84 // pStart is a pointer variable (see the * after the "char" type
85 // that points to a string of characters. This function prints
86 // each character in turn.
87 void Print1 (char* pStart)
88 {
89     // Loop until the end of string character is found
90     while(*pStart != '\0')
91     { // If not the end of string, print the character AND increment
92         // the pointer variable.
93         std::cout << *pStart++;
94     }
95 }
96
97 int main(int argc, char** argv)
98 { // argc is the count of the number of command line arguments
99     // and argv is the pointer to the array of arguments
100
101     // Use Print1 to print each argument
102     for (int i = 0; i < argc; i++)
103     {
104         Print1(argv[i]); // Note the function call with no return value
105         std::cout << '\n'; // and print the end of line
106     }
107
108     // Call the PriceModel function for several models
109     std::cout << "Cost of Ford is " << PriceModel(Ford) << std::endl;
110     std::cout << "Cost of Honda is " << PriceModel(Honda) << std::endl;
111     std::cout << "Cost of Mercedes is " << PriceModel(Mercedes) << std::endl;
112     std::cout << "Cost of Ferrari is " << PriceModel(Ferrari) << std::endl;

```

Program csyntax1.cc (continued)

```

113
114 // Illustrate the if/else construct
115 double hondaCost = PriceModel(Honda);
116 if (hondaCost > 15000)
117     { // Cost is more than $15000
118         std::cout << "Oops, the honda is too expensive" << std::endl;
119     }
120 else
121     { // Buy it
122         std::cout << "Ok, I'll take it" << std::endl;
123     }
124 // In the above, the "else" is optional. Often we have an "if" statement
125 // that takes some action if something is true, and no action if not.
126
127 // Illustrate the "sizeof" operator
128 std::cout << "sizeof int is " << sizeof(int) << std::endl;
129 std::cout << "sizeof char is " << sizeof(char) << std::endl;
130 std::cout << "sizeof float is " << sizeof(float) << std::endl;
131 std::cout << "sizeof double is " << sizeof(double) << std::endl;
132 std::cout << "sizeof myStruct is " << sizeof(myStruct) << std::endl;
133 }
134
135
136

```

Program csyntax1.cc (continued)