

Case Study: High Adventure Travel Agency—Part 1

The High Adventure Travel Agency offers four vacation packages for thrill-seeking customers. The rates and options vary for each package. You've been asked to write a program to calculate and itemize the charges for each package, described as follows:

Devil's Courthouse Adventure Weekend: An action-packed three-day weekend spent camping, rock climbing, and rappelling at Devil's Courthouse, North Carolina. This getaway is for novices and experts alike. Optional climbing instruction is available to beginners at a low price. Camping equipment rental is also available.

Rates

Base Charge:	\$350 per person
Climbing Instruction:	\$100 per person
Equipment Rental:	\$40/day per person

Scuba Bahama: A week-long cruise to the Bahamas with 3 days of scuba diving. Those with prior experience may dive right in, while beginners should choose to take optional, but very affordable lessons.

Rates

Base Charge:	\$1000 per person
Scuba Instruction:	\$100 per person

Sky Dive Colorado: Four thrilling days with expert sky-diving instructors in Colorado Springs, Colorado. For lodging, you may choose either the Wilderness Lodge or the Luxury Inn. (Instruction is included for all members of the party.)

Rates

Base Charge:	\$400 per person
Lodging at Wilderness Lodge:	\$65/day per person
Lodging at Luxury Inn:	\$120/day per person

Barron Cliff Spelunk: Eight days spent hiking and exploring caves in the Barron Cliff Wilderness Area, Tennessee. Camping equipment rental is available.

Rates

Base Charge:	\$700 per person
Equipment Rental:	\$40/day per person



NOTE: A 10 percent discount on the base charges of any package is given for a party of five or more.

Variables

Table 1 lists constant variables that will be defined globally. All these variables will be defined near the top of the program to make modifications easier (in the event the rates change).

Table 1 Global constants

Variable	Description
CLIMB_RATE = 350.0	A double. Holds base rate of Devil’s Courthouse Adventure Weekend package.
SCUBA_RATE = 1000.0	A double. Holds base rate of Scuba Bahamas package.
SKY_DIVE_RATE = 400.0	A double. Holds base rate of Sky Dive Colorado package.
CAVE_RATE = 700.0	A double. Holds base rate of Barron Cliff Spelunk package.
CLIMB_INSTRUCT = 100.0	A double. Holds charge for rock climbing instruction.
SCUBA_INSTRUCT = 100.0	A double. Holds charge for scuba instruction.
DAILY_CAMP_RENTAL = 40.0	A double. Holds daily charge, per person, for camping equipment rental.
DAY_LODGE_1 = 65.0	A double. Holds daily cost of lodging option 1 of Sky Dive Colorado package. (Wilderness Lodge.)
DAY_LODGE_2 = 120.0	A double. Holds daily cost of lodging option 2 of Sky Dive Colorado package. (Luxury Inn.)

Functions

Table 2 lists the functions used in the program.

Table 2 Functions

Function Name	Description
main	Calls the menu function and dispatches program control to the appropriate function, based on the user’s choice of packages.
menu	Displays a menu listing the vacation packages. Allows the user to enter a selection, which is returned to the main function.
climbing	Asks the user for data necessary to calculate charges for the Devil’s Courthouse Adventure Weekend package.
scuba	Asks the user for data necessary to calculate charges for the Scuba Bahamas package.
skyDive	Asks the user for data necessary to calculate charges for the Sky Dive Colorado package.
spelunk	Asks the user for data necessary to calculate charges for the Barron Cliff Spelunk package.

Program Design

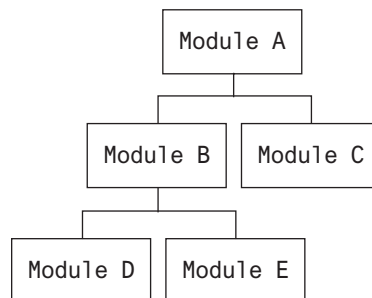
In modular programs, hierarchy charts are used to show the relationship between modules, or functions. For example, assume an application has five modules: Module A, Module B, Module C, Module D, and Module E. The relationship between the modules is

Module A calls Module B and Module C.

Module B calls Module D and Module E.

These relationships are shown in the hierarchy chart in Figure 1.

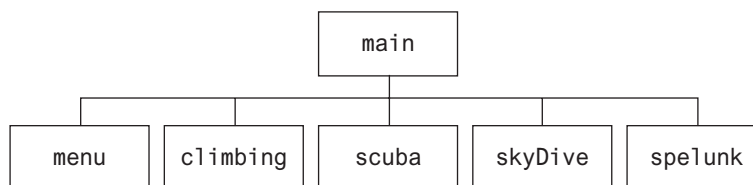
Figure 1 Hierarchy chart



Notice the hierarchy chart does not reveal details about the algorithm or specify when the modules are to be called. Instead, it reveals the relationship between the modules.

Figure 2 shows a hierarchy chart for the High Adventure Travel Agency program.

Figure 2 Hierarchy chart for the High Adventure Travel Agency program



The menu that is displayed should list the four vacation packages, plus a fifth option that exits the program. When the user chooses one of the vacation packages, a function is called that asks questions, such as how many people will be going on the trip, and calculates the charges for that particular package. This process repeats until the user chooses to exit the program.

Here is the pseudocode for each of the program's functions:

```
Main Function
  Call Menu Function
  Do While user does not enter 5
    Switch (User Input)
      Case 1: Call Climbing Function.
      Case 2: Call Scuba Function.
```

```

        Case 3: Call SkyDive Function.
        Case 4: Call Spelunk Function.
    End Switch.
End Do-While.
End of Main Function.

Climbing Function:
    Input number in party needing instruction.
    Input number of advanced climbers in party.
    Input number in party renting equipment.
    Calculate base charges.
    Calculate discount.
    Calculate cost of instruction.
    Calculate cost of equipment rental.
    Calculate total charges.
    Calculate minimum required deposit.
    Display results.
End of Climbing Function.

Scuba Function:
    Input number in party needing instruction.
    Input number of advanced scuba divers in party.
    Calculate base charges.
    Calculate discount.
    Calculate cost of instruction.
    Calculate total charges.
    Calculate minimum required deposit.
    Display results.
End of Scuba Function.

SkyDive Function:
    Input number in party.
    Calculate base charges.
    Calculate discount.
    Input number in party staying at Wilderness Lodge.
    Input number in party staying at Luxury Inn.
    Calculate lodging charges.
    Calculate total charges.
    Calculate minimum required deposit.
    Display results.
End of SkyDive Function.

Spelunk Function:
    Input number in party.
    Input number in party renting equipment.
    Calculate base charges.
    Calculate discount.
    Calculate cost of equipment rental.
    Calculate total charges.
    Calculate minimum required deposit.
    Display results.
End of Spelunk Function.

```

Program CS2-1 lists the C++ code.

Program CS2-1

```

1 // This program will assist the High Adventure Travel Agency
2 // in calculating the costs of their 4 major vacation packages.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 // Constants for the charges.
8 const double CLIMB_RATE = 350.0;           // Base rate - Devil's Courthouse
9 const double SCUBA_RATE = 1000.0;         // Base rate - Bahamas
10 const double SKY_DIVE_RATE = 400.0;       // Base rate - Sky diving
11 const double CAVE_RATE = 700.0;          // Base rate - Spelunking
12 const double CLIMB_INSTRUCT = 100.0;      // Climbing instruction
13 const double SCUBA_INSTRUCT = 100.0;      // Scuba instruction
14 const double DAILY_CAMP_RENTAL = 40.0;     // Daily camping equipment rental
15 const double DAY_LODGE_1 = 65.0;          // Lodging option (sky diving)
16 const double DAY_LODGE_2 = 120.0;         // Lodging option (sky diving)
17
18 // Function prototypes
19 void climbing();
20 void scuba();
21 void skyDive();
22 void spelunk();
23 int menu();
24
25 int main()
26 {
27     int selection;
28
29     cout << fixed << showpoint << setprecision(2);
30     do
31     {
32         selection = menu();
33         switch(selection)
34         {
35             case 1 : climbing();
36                     break;
37             case 2 : scuba();
38                     break;
39             case 3 : skyDive();
40                     break;
41             case 4 : spelunk();
42                     break;
43             case 5 : cout << "Exiting program.\n\n";
44                     }
45         } while (selection != 5);
46         return 0;
47     }
48
49 // *****
50 // Definition of function menu. *
51 // Displays the main menu and asks the user to select *
52 // an option. Returns an integer in the range 1 – 5. *
53 // *****

```

(program continues)

Program CS2-1 (continued)

```

54
55 int menu()
56 {
57     int choice;
58
59     cout << "High Adventure Travel Agency\n";
60     cout << "-----\n";
61     cout << "1) Devil's Courthouse Adventure Weekend\n";
62     cout << "2) Scuba Bahama\n";
63     cout << "3) Sky Dive Colorado\n";
64     cout << "4) Barron Cliff Spelunk\n";
65     cout << "5) Exit Program\n\n";
66     cout << "Enter 1, 2, 3, 4, or 5: ";
67     cin >> choice;
68     while (choice < 1 || choice > 5) // Validate input
69     {
70         cout << "Invalid Selection. Enter 1, 2, 3, 4, or 5: ";
71         cin >> choice;
72     }
73     return choice;
74 }
75
76 // ***** This work is protected by United States copyright laws
77 // ***** and is provided solely for the use of instructors in teaching
78 // ***** their courses and assessing student learning. ***** Dissemination
79 // ***** of this work or its contents in any form (print or electronic) without the publisher's
80 // ***** written permission is prohibited. ***** (World Wide Web)
81
82 void climbing()
83 {
84     int    beginners,    // Those needing instruction
85           advanced,      // Those not needing instruction
86           needEquip;     // Those renting camping equipment
87     double baseCharges, // Base charges
88           charges,       // Total charges
89           instruction,   // Cost of instruction
90           equipment,     // Cost of equipment rental
91           discount = 0,   // Discount
92           deposit;       // Required deposit
93
94     cout << "\nDevil's Courthouse Adventure Weekend\n";
95     cout << "-----\n";
96     cout << "How many will be going who need an instructor? ";
97     cin >> beginners;
98     cout << "How many advanced climbers will be going? ";
99     cin >> advanced;
100    cout << "How many will rent camping equipment? ";
101    cin >> needEquip;
102    // Calculate base charges.
103    baseCharges = (beginners + advanced) * CLIMB_RATE;
104    charges = baseCharges;

```

```

105 // Calculate 10% discount for 5 or more.
106 if ((beginners + advanced) > 4)
107 {
108     discount = (charges * .1);
109     charges -= discount;
110 }
111 // Add cost of instruction.
112 instruction = beginners * CLIMB_INSTRUCT;
113 charges += instruction;
114 // Add cost of camping equipment rental
115 equipment = needEquip * DAILY_CAMP_RENTAL * 4;
116 charges += equipment;
117 // Calculate required deposit.
118 deposit = charges / 2.0;
119 cout << "Number in party: " << (beginners + advanced);
120 cout << endl;
121 cout << "Base charges: $" << baseCharges << endl;
122 cout << "Instruction cost: $" << instruction << endl;
123 cout << "Equipment Rental: $" << equipment << endl;
124 cout << "Discount: $" << discount << endl;
125 cout << "Total Charges: $" << charges << endl;
126 cout << "Required Deposit: $" << deposit << endl << endl;
127 }
128
129 // *****
130 // Definition of scuba function.
131 // This function calculates the charges for the
132 // Scuba Bahama package.
133 // *****
134
135 void scuba()
136 {
137     int    beginners,    // Those needing instruction
138           advanced;      // Those not needing instruction
139     double baseCharges,  // Base charges
140           charges,       // Total charges
141           instruction,   // Cost of instruction
142           discount = 0,  // Discount
143           deposit;       // Required deposit
144
145     cout << "\nScuba Bahama\n";
146     cout << "-----\n";
147     cout << "How many will be going who need an instructor? ";
148     cin >> beginners;
149     cout << "How many advanced scuba divers will be going? ";
150     cin >> advanced;
151     // Calculate base charges.
152     baseCharges = (beginners + advanced) * SCUBA_RATE;
153     charges = baseCharges;

```

(program continues)

Program CS2-1 (continued)

```

154     // Calculate 10% discount for 5 or more.
155     if ((beginners + advanced) > 4)
156     {
157         discount = (charges * .1);
158         charges -= discount;
159     }
160     // Add cost of instruction.
161     instruction = beginners * SCUBA_INSTRUCT;
162     charges += instruction;
163
164     // Calculate required deposit.
165     deposit = charges / 2.0;
166     cout << "Number in party: " << (beginners + advanced);
167     cout << endl;
168     cout << "Base charges: $" << baseCharges << endl;
169     cout << "Instruction cost: $" << instruction << endl;
170     cout << "Discount: $" << discount << endl;
171     cout << "Total Charges: $" << charges << endl;
172     cout << "Required Deposit: $" << deposit << endl << endl;
173 }
174
175 // *****
176 // Definition of skyDive function.
177 // This function calculates the charges for the
178 // Sky Dive Colorado package.
179 // *****
180
181 void skyDive()
182 {
183     int    party,           // Number in party
184           lodge1,          // Number at 1st lodging choice
185           lodge2;          // Number at 2nd lodging choice
186     double baseCharges,    // Base charges
187           charges,         // Total charges
188           discount = 0,    // Discount
189           lodging,         // Cost of lodging
190           deposit;         // Required deposit
191
192     cout << "\nSky Dive Colorado\n";
193     cout << "-----\n";
194     cout << "How many will be going? ";
195     cin >> party;
196     // Calculate base charges.
197     baseCharges = party * SKY_DIVE_RATE;
198     charges = baseCharges;
199     // Calculate 10% discount for 5 or more.
200     if (party > 4)
201     {
202         discount = (charges * .1);
203         charges -= discount;
204     }

```



```

205 // Calculate lodging costs.
206 cout << "How many will stay at Wilderness Lodge? ";
207 cin >> lodge1;
208 cout << "How many will stay at Luxury Inn? ";
209 cin >> lodge2;
210 lodging = (lodge1 * DAY_LODGE_1) + (lodge2 * DAY_LODGE_2);
211 charges += lodging;
212
213 // Calculate required deposit.
214 deposit = charges / 2.0;
215 cout << "Number in party: " << party << endl;
216 cout << "Base charges: $" << baseCharges << endl;
217 cout << "Lodging: $" << lodging << endl;
218 cout << "Discount: $" << discount << endl;
219 cout << "Total Charges: $" << charges << endl;
220 cout << "Required Deposit: $" << deposit << endl << endl;
221 }
222
223 //*****
224 // Definition of spelunk function. *
225 // This function calculates the charges for the *
226 // Barron Cliff Spelunk package. *
227 //*****
228
229 void spelunk()
230 {
231     int party, // Number in party
232     needEquip; // Those renting camping equipment
233     double baseCharges, // Base charges
234     charges, // Total charges
235     equipment, // Cost of equipment rental
236     discount = 0, // Discount
237     deposit; // Required deposit
238
239     cout << "\nBarron Cliff Spelunk Weekend\n";
240     cout << "-----\n";
241     cout << "How many will be going? ";
242     cin >> party;
243     cout << "How many will rent camping equipment? ";
244     cin >> needEquip;
245     // Calculate base charges.
246     baseCharges = party * CAVE_RATE;
247     charges = baseCharges;
248     // Calculate 10% discount for 5 or more.
249     if (party > 4)
250     {
251         discount = (charges * .1);
252         charges -= discount;
253     }
254     // Add cost of camping equipment rental
255     equipment = needEquip * DAILY_CAMP_RENTAL * 4;
256     charges += equipment;
257

```

(program continues)

Program CS2-1 (continued)

```

258     // Calculate required deposit.
259     deposit = charges / 2.0;
260     cout << "Number in party: " << party << endl;
261     cout << "Base charges: $" << baseCharges << endl;
262     cout << "Equipment Rental: $" << equipment << endl;
263     cout << "Discount: $" << discount << endl;
264     cout << "Total Charges: $" << charges << endl;
265     cout << "Required Deposit: $" << deposit << endl << endl;
266 }

```

Program Output with Example Input Shown in Bold

High Adventure Travel Agency

- 1) Devil's Courthouse Adventure Weekend
- 2) Scuba Bahama
- 3) Sky Dive Colorado
- 4) Barron Cliff Spelunk
- 5) Exit Program

Enter 1, 2, 3, 4, or 5: **1**

Devil's Courthouse Adventure Weekend

How many will be going who need an instructor? **3**

How many advanced climbers will be going? **2**

How many will rent camping equipment? **3**

Number in party: 5

Base charges: \$1750.00

Instruction cost: \$300.00

Equipment Rental: \$480.00

Discount: \$175.00

Total Charges: \$2355.00

Required Deposit: \$1177.50

High Adventure Travel Agency

- 1) Devil's Courthouse Adventure Weekend
- 2) Scuba Bahama
- 3) Sky Dive Colorado
- 4) Barron Cliff Spelunk
- 5) Exit Program

Enter 1, 2, 3, 4, or 5: **2**

Scuba Bahama

How many will be going who need an instructor? 4 How many advanced scuba divers will be going? 0

Number in party: 4

Base charges: \$4000.00

Instruction cost: \$400.00

Discount: \$0.00

Total Charges: \$4400.00

Required Deposit: \$2200.00

High Adventure Travel Agency

- 1) Devil's Courthouse Adventure Weekend
- 2) Scuba Bahama
- 3) Sky Dive Colorado
- 4) Barron Cliff Spelunk
- 5) Exit Program

Enter 1, 2, 3, 4, or 5: 3

Sky Dive Colorado

How many will be going? 8 How many will stay at Wilderness Lodge? 4 How many will stay at Luxury Inn? 4

Number in party: 8

Base charges: \$3200.00

Lodging: \$740.00

Discount: \$320.00

Total Charges: \$3620.00

Required Deposit: \$1810.00

High Adventure Travel Agency

- 1) Devil's Courthouse Adventure Weekend
- 2) Scuba Bahama
- 3) Sky Dive Colorado
- 4) Barron Cliff Spelunk
- 5) Exit Program

Enter 1, 2, 3, 4, or 5: 4 *(program output continues)*

Program CS2-1 (continued)

Barron Cliff Spelunk Weekend

How many will be going? **6**

How many will rent camping equipment? **2**

Number in party: 6

Base charges: \$4200.00

Equipment Rental: \$320.00

Discount: \$420.00

Total Charges: \$4100.00

Required Deposit: \$2050.00

High Adventure Travel Agency

- 1) Devil's Courthouse Adventure Weekend
- 2) Scuba Bahama
- 3) Sky Dive Colorado
- 4) Barron Cliff Spelunk
- 5) Exit Program

Enter 1, 2, 3, 4, or 5: **5**

Exiting program.

This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted.