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 rapelling 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

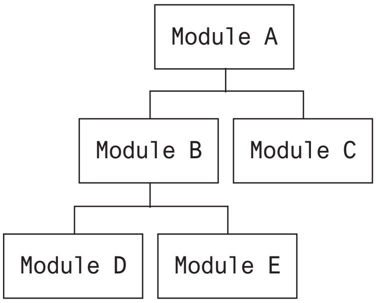
In modular programs, hierarchy charts are used to show the relationship between modules, or functions. For example, assume an application has five modules: Module A, Module B, Module C, Module D, and Module E. The relationship between the modules is

`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.

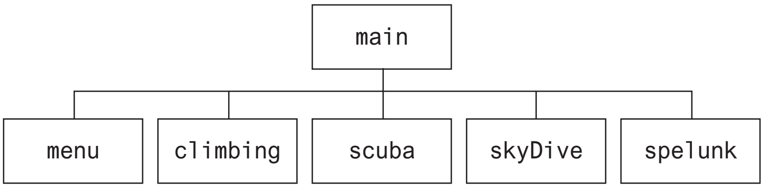
Figure 1 Hierarchy chart



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

```
// This program will assist the High Adventure Travel Agency
// in calculating the costs of their 4 major vacation packages.
#include <iostream>
#include <iomanip>
using namespace std;

// Constants for the charges.
const double CLIMB_RATE = 350.0;       // Base rate - Devil's Courthouse
const double SCUBA_RATE = 1000.0;      // Base rate - Bahamas
const double SKY_DIVE_RATE = 400.0;    // Base rate - Sky diving
const double CAVE_RATE = 700.0;        // Base rate - Spelunking
const double CLIMB_INSTRUCT = 100.0;   // Climbing instruction
const double SCUBA_INSTRUCT = 100.0;   // Scuba instruction
const double DAILY_CAMP_RENTAL = 40.0; // Daily camping equipment rental
const double DAY_LODGE_1 = 65.0;       // Lodging option (sky diving)
const double DAY_LODGE_2 = 120.0;      // Lodging option (sky diving)

// Function prototypes
void climbing();
void scuba();
void skyDive();
void spelunk();
int menu();

int main()
{
    int selection;

    cout << fixed << showpoint << setprecision(2);
    do
    {
        selection = menu();
        switch(selection)
        {
            case 1 : climbing();
                     break;
            case 2 : scuba();
                     break;
            case 3 : skyDive();
                     break;
            case 4 : spelunk();
                     break;
            case 5 : cout << "Exiting program.\n\n";
        }
    } while (selection != 5);
    return 0;
}

//******************************************************
// Definition of function menu.                        *
// Displays the main menu and asks the user to select  *
// an option. Returns an integer in the range 1 – 5.   *
//******************************************************
```

(program continues)

Program CS2-1 *(continued)*

```

int menu()
{
    int choice;

    cout << "High Adventure Travel Agency\n";
    cout << "---------------------------\n";
    cout << "1) Devil's Courthouse Adventure Weekend\n";
    cout << "2) Scuba Bahama\n";
    cout << "3) Sky Dive Colorado\n";
    cout << "4) Barron Cliff Spelunk\n";
    cout << "5) Exit Program\n\n";
    cout << "Enter 1, 2, 3, 4, or 5: ";
    cin >> choice;
    while (choice < 1 || choice > 5) // Validate input
    {
        cout << "Invalid Selection. Enter 1, 2, 3, 4, or 5: ";
        cin >> choice;
    }
    return choice;
}

//*************************************************
// Definition of climbing function.               *
// This function calculates the charges for the   *
// Devil's Courthouse Adventure Weekend package.  *
//*************************************************

void climbing()
{
    int  beginners,      // Those needing instruction
         advanced,       // Those not needing instruction
         needEquip;      // Those renting camping equipment
    double baseCharges,  // Base charges
         charges,        // Total charges
         instruction,    // Cost of instruction
         equipment,      // Cost of equipment rental
         discount = 0,   // Discount
         deposit;        // Required deposit

    cout << "\nDevil's Courthouse Adventure Weekend\n";
    cout << "-----------------------------------\n";
    cout << "How many will be going who need an instructor? ";
    cin >> beginners;
    cout << "How many advanced climbers will be going? ";
    cin >> advanced;
    cout << "How many will rent camping equipment? ";
    cin >> needEquip;
    // Calculate base charges.
    baseCharges = (beginners + advanced) * CLIMB_RATE;
    charges = baseCharges;
    // Calculate 10% discount for 5 or more.
    if ((beginners + advanced) > 4)
    {
        discount = (charges * .1);
        charges -= discount;
    }
    // Add cost of instruction.
    instruction = beginners * CLIMB_INSTRUCT;
    charges += instruction;
    // Add cost of camping equipment rental
    equipment = needEquip * DAILY_CAMP_RENTAL * 4;
    charges += equipment;
    // Calculate required deposit.
    deposit = charges / 2.0;
    cout << "Number in party: " << (beginners + advanced);
    cout << endl;
    cout << "Base charges: $" << baseCharges << endl;
    cout << "Instruction cost: $" << instruction << endl;
    cout << "Equipment Rental: $" << equipment << endl;
    cout << "Discount: $" << discount << endl;
    cout << "Total Charges: $" << charges << endl;
    cout << "Required Deposit: $" << deposit << endl << endl;
}

//************************************************
// Definition of scuba function.                *
// This function calculates the charges for the  *
// Scuba Bahama package.                         *
//************************************************

void scuba()
{
    int    beginners,     // Those needing instruction
           advanced;      // Those not needing instruction
    double baseCharges,   // Base charges
           charges,       // Total charges
           instruction,   // Cost of instruction
           discount = 0,  // Discount
           deposit;       // Required deposit

    cout << "\nScuba Bahama\n";
    cout << "------------------------------------\n";
    cout << "How many will be going who need an instructor? ";
    cin >> beginners;
    cout << "How many advanced scuba divers will be going? ";
    cin >> advanced;
    // Calculate base charges.
    baseCharges = (beginners + advanced) * SCUBA_RATE;
    charges = baseCharges;
```

(program continues)

Program CS2-1 *(continued)*

```
    // Calculate 10% discount for 5 or more.
    if ((beginners + advanced) > 4)
    {
        discount = (charges * .1);
        charges -= discount;
    }
    // Add cost of instruction.
    instruction = beginners * SCUBA_INSTRUCT;
    charges += instruction;

    // Calculate required deposit.
    deposit = charges / 2.0;
    cout << "Number in party: " << (beginners + advanced);
    cout << endl;
    cout << "Base charges: $" << baseCharges << endl;
    cout << "Instruction cost: $" << instruction << endl;
    cout << "Discount: $" << discount << endl;
    cout << "Total Charges: $" << charges << endl;
    cout << "Required Deposit: $" << deposit << endl << endl;
}

//************************************************
// Definition of skyDive function.               *
// This function calculates the charges for the  *
// Sky Dive Colorado package.                    *
//************************************************

void skyDive()
{
    int    party,          // Number in party
           lodge1,         // Number at 1st lodging choice
           lodge2;         // Number at 2nd lodging choice
    double baseCharges,    // Base charges
           charges,        // Total charges
           discount = 0,   // Discount
           lodging,        // Cost of lodging
           deposit;        // Required deposit

    cout << "\nSky Dive Colorado\n";
    cout << "------------------------------------\n";
    cout << "How many will be going? ";
    cin >> party;
    // Calculate base charges.
    baseCharges = party * SKY_DIVE_RATE;
    charges = baseCharges;
    // Calculate 10% discount for 5 or more.
    if (party > 4)
    {
        discount = (charges * .1);
        charges -= discount;
    }
    // Calculate lodging costs.
    cout << "How many will stay at Wilderness Lodge? ";
    cin >> lodge1;
    cout << "How many will stay at Luxury Inn? ";
    cin >> lodge2;
    lodging = (lodge1 * DAY_LODGE_1) + (lodge2 * DAY_LODGE_2);
    charges += lodging;

    // Calculate required deposit.
    deposit = charges / 2.0;
    cout << "Number in party: " << party << endl;
    cout << "Base charges: $" << baseCharges << endl;
    cout << "Lodging: $" << lodging << endl;
    cout << "Discount: $" << discount << endl;
    cout << "Total Charges: $" << charges << endl;
    cout << "Required Deposit: $" << deposit << endl << endl;
}

//************************************************
// Definition of spelunk function.               *
// This function calculates the charges for the  *
// Barron Cliff Spelunk package.                 *
//************************************************

void spelunk()
{
    int    party,          // Number in party
           needEquip;      // Those renting camping equipment
    double baseCharges,    // Base charges
           charges,        // Total charges
           equipment,      // Cost of equipment rental
           discount = 0,   // Discount
           deposit;        // Required deposit

    cout << "\nBarron Cliff Spelunk Weekend\n";
    cout << "------------------------------------\n";
    cout << "How many will be going? ";
    cin >> party;
    cout << "How many will rent camping equipment? ";
    cin >> needEquip;
    // Calculate base charges.
    baseCharges = party * CAVE_RATE;
    charges = baseCharges;
    // Calculate 10% discount for 5 or more.
    if (party > 4)
    {
        discount = (charges * .1);
        charges -= discount;
    }
    // Add cost of camping equipment rental
    equipment = needEquip * DAILY_CAMP_RENTAL * 4;
    charges += equipment;

```

(program continues)

Program CS2-1 *(continued)*

```
    // Calculate required deposit.
    deposit = charges / 2.0;
    cout << "Number in party: " << party << endl;
    cout << "Base charges: $" << baseCharges << endl;
    cout << "Equipment Rental: $" << equipment << endl;
    cout << "Discount: $" << discount << endl;
    cout << "Total Charges: $" << charges << endl;
    cout << "Required Deposit: $" << deposit << endl << endl;
}
```

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 [Enter]

Devil's Courthouse Adventure Weekend
------------------------------------
How many will be going who need an instructor? 3 [Enter]
How many advanced climbers will be going? 2 [Enter]
How many will rent camping equipment? 3 [Enter]
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 [Enter]
```

```
Scuba Bahama
------------------------------------
How many will be going who need an instructor? 4 Enter
How many advanced scuba divers will be going? 0 Enter
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 Enter

Sky Dive Colorado
------------------------------------
How many will be going? 8 Enter
How many will stay at Wilderness Lodge? 4 Enter
How many will stay at Luxury Inn? 4 Enter
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 Enter
```

(program output continues)

Program CS2-1 *(continued)*

```
Barron Cliff Spelunk Weekend
------------------------------------
How many will be going? 6 Enter
How many will rent camping equipment? 2 Enter
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 Enter
Exiting program.
```