

```

//
//  main.cpp
//  AbsoluteCpp_ch4_2
//

//Program to demonstrate call-by-reference parameters.
#include <iostream>
#include <cstdlib>
#include <fstream>
using namespace std;

void getNumbers(int& input1, int& input2);
//Reads two integers from the keyboard.

void swapValues(int& variable1, int& variable2);
//Interchanges the values of variable1 and variable2.

void showResults(int output1, int output2);
//Shows the values of variable1 and variable2, in that order.

int main( )
{
    int firstNum, secondNum;

    getNumbers(firstNum, secondNum);
    swapValues(firstNum, secondNum);
    showResults(firstNum, secondNum);
    return 0;
}

void getNumbers(int& input1, int& input2)
{
    cout << "Enter two integers: ";
    cin >> input1
        >> input2;
}

void swapValues(int& variable1, int& variable2)
{
    int temp;

    temp = variable1;
    variable1 = variable2;
    variable2 = temp;
}

void showResults(int output1, int output2)
{
    cout << "In reverse order the numbers are: "
        << output1 << " " << output2 << endl;
}

```

```

//
//  main.cpp
//  AbsoluteCpp_ch4_3
//

//Illustrates the difference between a call-by-value
//parameter and a call-by-reference parameter.
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cmath>
using namespace std;

void doStuff(int par1Value, int& par2Ref);
//par1Value is a call-by-value formal parameter and
//par2Ref is a call-by-reference formal parameter.

int main( )
{
    int n1, n2;

    n1 = 1;
    n2 = 2;
    doStuff(n1, n2);
    cout << "n1 after function call = " << n1 << endl;
    cout << "n2 after function call = " << n2 << endl;
    return 0;
}

void doStuff(int par1Value, int& par2Ref)
{
    par1Value = 111;
    cout << "par1Value in function call = "
         << par1Value << endl;
    par2Ref = 222;
    cout << "par2Ref in function call = "
         << par2Ref << endl;
}

```

```

//
//  main.cpp
//  AbsoluteCpp_ch4_5
//

//Determines which of two pizza sizes is the best buy.
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <cmath>
using namespace std;

void getData(int& smallDiameter, double& priceSmall,
            int& largeDiameter, double& priceLarge);

void giveResults(int smallDiameter, double priceSmall,
                int largeDiameter, double priceLarge);

double unitPrice(int diameter, double price);
//Returns the price per square inch of a pizza.
//Precondition: The diameter parameter is the diameter of the pizza
//in inches. The price parameter is the price of the pizza.

int main( )
{
    int diameterSmall, diameterLarge;
    double priceSmall, priceLarge;

    getData(diameterSmall, priceSmall, diameterLarge, priceLarge);
    giveResults(diameterSmall, priceSmall, diameterLarge, priceLarge);

    return 0;
}

void getData(int& smallDiameter, double& priceSmall,
            int& largeDiameter, double& priceLarge)
{
    cout << "Welcome to the Pizza Consumers Union.\n";
    cout << "Enter diameter of a small pizza (in inches): ";
    cin >> smallDiameter;
    cout << "Enter the price of a small pizza: $";
    cin >> priceSmall;
    cout << "Enter diameter of a large pizza (in inches): ";
    cin >> largeDiameter;
    cout << "Enter the price of a large pizza: $";
    cin >> priceLarge;
}

void giveResults(int smallDiameter, double priceSmall,
                int largeDiameter, double priceLarge)
{
    double unitPriceSmall, unitPriceLarge;

    unitPriceSmall = unitPrice(smallDiameter, priceSmall);

```

```

unitPriceLarge = unitPrice(largeDiameter, priceLarge);

cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);
cout << "Small pizza:\n"
    << "Diameter = " << smallDiameter << " inches\n"
    << "Price = $" << priceSmall
    << " Per square inch = $" << unitPriceSmall << endl
    << "Large pizza:\n"
    << "Diameter = " << largeDiameter << " inches\n"
    << "Price = $" << priceLarge
    << " Per square inch = $" << unitPriceLarge << endl;
if (unitPriceLarge < unitPriceSmall)
    cout << "The large one is the better buy.\n";
else
    cout << "The small one is the better buy.\n";
cout << "Buon Appetito!\n";
}

double unitPrice(int diameter, double price)
{
    const double PI = 3.14159;
    double radius, area;

    radius = diameter/static_cast<double>(2);
    area = PI * radius * radius;
    return (price/area);
}

```

```

//
//  main.cpp
//  AbsoluteCpp_ch4_8
//

#include <iostream>
using namespace std;

void showVolume(int length, int width = 1, int height = 1);
void showVolume(int length);
//Returns the volume of a box.
//If no height is given, the height is assumed to be 1.
//If neither height nor width are given, both are assumed to be 1.

int main( )
{
    showVolume(4, 6, 2);
    showVolume(4, 6);

    // This calling causes a conflict, since there are 2 functions named
    // showVolume
    // with 1 argument.

    //showVolume(4);

    return 0;
}

void showVolume(int length, int width, int height)
{
    cout << "Volume of a box with \n"
         << "Length = " << length << ", Width = " << width << endl
         << "and Height = " << height
         << " is " << length*width*height << endl;
}

void showVolume(int length){
    cout << "Volume of cube with \n" << "Length = " << length <<endl << "
         is " <<
         length * length * length << endl;
}

```

```

//
//  main.cpp
//  AbsoluteCpp_ch4_9
//

//Driver program for the function unitPrice.
#include <iostream>
using namespace std;

double unitPrice(int diameter, double price);
//Returns the price per square inch of a pizza.
//Precondition: The diameter parameter is the diameter of the pizza
//in inches. The price parameter is the price of the pizza.

int main( )
{
    double diameter, price;
    char ans;

    do
    {
        cout << "Enter diameter and price:\n";
        cin >> diameter >> price;

        cout << "unit Price is $"
             << unitPrice(diameter, price) << endl;

        cout << "Test again? (y/n)";
        cin >> ans;
        cout << endl;
    } while (ans == 'y' || ans == 'Y');

    return 0;
}

double unitPrice(int diameter, double price)
{
    const double PI = 3.14159;
    double radius, area;

    radius = diameter/static_cast<double>(2);
    area = PI * radius * radius;
    return (price/area);
}

```