**//Functions3.cpp**

**// Demonstrates the use of pass by reference**

**//Author: nmessa**

**//Date: 2/8/2016**

#include <iostream>

using namespace std;

//Function Prototype

void add(int, int, int &);

int main()

{

//Declare variables

int number1, number2, sum;

//Input two integers

cout << "Enter a number: ";

cin >> number1;

cout << "Enter a number: ";

cin >> number2;

//Send numbers to function and which calculates

//and "returns" the answer by reference

add(number1, number2, sum);

//Output the answer

cout << number1 << " + " << number2 << " = " << sum << endl;

return 0;

}

//Function add returns the sum of the two parameters that

//are sent to it with the answer "returned" as a

//reference parameter

void add(int x, int y, int &answer)

{

answer = x + y;

}

**//Functions4.cpp**

**//Author: nmessa**

**//Date: 2/8/2016**

**//This program demonstrates the use of function overloading**

#include <iostream>

#include <cmath>

using namespace std;

//define a global constant

const double PI = 4\*atan(1.0);

//Function prototypes

double area(double);

double area(double, double);

int main()

{

//Declare variables

double radius, length, width, circleArea, rectangleArea;

//Find area of a circle and print out result

cout << "Area of Circle" << endl;

cout << "Enter the radius: ";

cin >> radius;

circleArea = area(radius);

cout << "Circle Area = " << circleArea << endl;

cout << "\n\n\n";

//Find area of a rectangle and print out result

cout << "Area of Rectangle" << endl;

cout << "Enter the length: ";

cin >> length;

cout << "Enter the width: ";

cin >> width;

rectangleArea = area(length, width);

cout << "Rectangle Area = " << rectangleArea << endl;

return 0;

}

//Returns the area of a circle with a radius of r

double area(double r)

{

double answer;

answer = 2 \* PI \* pow(r, 2);

return answer;

}

//Returns the area of a rectangle of length l and

//width w

double area(double l, double w)

{

double answer;

answer = l \* w;

return answer;

}

**//Function Demo Program**

**//Author: nmessa**

**//Date: 2/8/2016**

#include <iostream>

using namespace std;

//Function Prototypes

void swap(int, int);

void swap2(int &, int &);

void swap3(int\*, int\*);

int main()

{

int number1 = 42;

int number2 = 17;

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

cout << "Pass by value call" << endl;

swap(number1, number2);

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

cout << "\n\n\n";

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

cout << "Pass by reference call" << endl;

swap2(number1, number2);

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

cout << "\n\n\n";

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

cout << "Pass by pointer call" << endl;

swap3(&number1, &number2);

cout << "Number 1 = " << number1 << endl;

cout << "Number 2 = " << number2 << endl;

return 0;

}

//Pass by value swap function

void swap(int n1, int n2)

{

int temp;

temp = n1;

n1 = n2;

n2 = temp;

}

//Pass by reference swap function

void swap2(int &n1, int &n2)

{

int temp;

temp = n1;

n1 = n2;

n2 = temp;

}

//Pass by pointer swap function

void swap3(int\* n1, int \*n2)

{

int temp;

temp = \*n1;

\*n1 = \*n2;

\*n2 = temp;

}