To do for week of 3/17 – Function Overloading and Scope rules

Functions

Review pass by value

Pass by reference

Void getGrades (int &grade1, int &grade2, int &grade3)

Function overloading

A function could use different set of parameters by overload it.

In the following program see the three prototypes for with the same function name

----------------------

#include <iostream>

using namespace std;

double findAv(int, int, int);

double findAv(double,double,double);

double findAv(double,double,double,double);

int main ()

{

double average;

//call findAv with three integer parameters

average = findAv(80,82,95);

cout << "\n Called overloaded function with three integer parameters.

Average is: " <<average;

//call findAv with three double parameters

average = findAv(82.5,89.4,95.88);

cout << "\n Called overloaded function with three double parameters.

Average is: " <<average;

//call findAv with four double parameters

average = findAv(82.5,89.4,95.88, 99.89);

cout << "\n Called overloaded function with four double parameters.

Average is: " <<average;

return 0;

}

double findAv(int a, int b, int c)

{

return (a+b+c)/3.0;

}

double findAv(double a, double b, double c)

{

return (a+b+c)/3.0;

}

double findAv(double a, double b, double c, double d)

{

return (a+b+c+d)/4.0; ------------------------------------------------scope rules -----------------------

// scopeRulesFunctions.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int getDiscount();

int discount =50;

int \_tmain(int argc, \_TCHAR\* argv[])

{

cout << "global discount rage is : " <<discount<<"%\n";

int discount=20;

cout <<"Discount in the main is: "<<discount<<"%\n";

discount = getDiscount();

cout <<"Discount received from the function is: "<<discount<<"%\n";

cout <<"Finally the only discount visible in the main now is: "<<discount<<"%\n";

return 0;

}

int getDiscount()

{

int discount =90;

cout <<"Discount rate in the function is " <<discount<<"%\n";

cout <<"Enter special discount rate: ";cin>>discount;

return discount;

}

To do list for the week of 3/17

Complete functions. Specifically overloaded functions, value and reference parameters, and scope of variables.

Start on arrays –

**What is an array?** An array is used to store nd process a collection of data of the same type;

it cannot hold values of different types. An array may also be called a list.

**How is an array declared**? Suppose we want to work with grades for students in different classes. We have to decide what is the maximum number of scores we will expect to have. For now, we will choose 90. This is a limitation of arrays; we are limited to this number once we code it in. To change it we will have to change the code and recompile. That is why we call an array a static list. To declare an array

of 90 scores declare it like any other integer variable, but put a [90] in the square

brackets: int scores[90].

Just because we declared an array for 90 elements, there is no need to enter all 90

scores. We can enter any number of scores up to 90

Example of Reading scores where the array is declared globally.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accept some scores into an array.

Terminate Entry when a negative number is entered.

Keep track of number of scores entered.

Teaching objective - Data entry into an array.

By Dr. John Abraham

Created for 1370 students

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

int scores[90]; //array is declared globally for this example.

int getscores (void);

int main ()

{

int n, i; // n for number of scores. i is a counter.

n = getscores(); //go get the scores and how many

for (i=1; i<=n; i++) cout<<scores[i]<<endl; //show scores.

return(0);

}

int getscores()

{

int n=1;

cout << "ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY

ENTERING A NEG NUMBER!\n";

cout << "Enter score# " << n << " ";

cin >> scores[n];

while (scores[n] >= 0)

{

n++;

cout << "Enter score# " << n <<" ";

cin >> scores[n];

}

return n-1; //n-1 actual scores read.

}

Program Run 11-1.

ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A

NEG NUMBER!

Enter score# 1 78

Enter score# 2 91

Enter score# 3 88

Enter score# 4 75

Enter score# 5 60

Enter score# 6 88

Enter score# 7 59

Enter score# 8 99

Enter score# 9 78

Enter score# 10 -1

78

91

88

75

60

88 59

99

78

Press any key to continue

**How do we pass array as a parameter into a function?**  Passing array as a parameter is not all that difficult. First, include a prototype which indicates that the parameter passed is an array, without actually showing the number of elements in the array - just put []. The function heading also has a similar declaration.

This program is passing an array as a parameter.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accept some scores into an array.

Terminate Entry when a negative number is entered.

Keep track of number of scores entered.

Teaching objective - Pass Array as a parameter.

By Dr. John Abraham

Created for 1370 students

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

int getscores (int scores[]); //prototype for passing an array

int main ()

{

int scores[90]; //array of 90 integers named

scores

int n, i; // n for number of scores.

i is a counter.

n = getscores(scores); //go get the scores and how many

for (i=1; i<=n; i++) cout<<scores[i]<<endl; //show scores.

return(0);

}

int getscores(int scores[])

{

int n=1;

cout << "ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY

ENTERING A NEG NUMBER!\n";

cout << "Enter score# " << n << " ";

cin >> scores[n];

while (scores[n] >= 0)

{

n++; cout << "Enter score# " << n <<" ";

cin >> scores[n];

}

return n-1; //n-1 actual scores read.

}

Program Run 11-2.

ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A

NEG NUMBER!

Enter score# 1 88

Enter score# 2 77

Enter score# 3 76

Enter score# 4 -1

88

77

76

Press any key to continue

It is important to note that the array parameter passed to the function is neither a

call-by-value, or a call-by-reference, but a new kind of parameter known as an array

parameter. It follows that when passing an array neither a copy of the entire array is

passed nor every memory location of entire array are passed, instead the memory location

of the first element of the array is passed. Therefore, it is not necessary to pass the size of

the array, just include square brackets with nothing in it. Both the calling module and

called module calculate the memory location of a particular element using a base and

offset (calculated by index). More about addressing modes will be taught in future

courses