**ITCS 1212L**

**Lab 4**

**Introduction to Functions and Modular Programs**

**Learning Objectives:**

* **To learn the process of creating functions and function prototypes.**
* **To learn the process of calling a function.**
* **To learn how to create a modular program.**
* **Practice developing the main algorithm which is composed of different functions calls.**

1. Practice with function definition, function call and function prototype.

Create a project named lab4A.cbj in codeblocks and type the following program to print your name and your lab partner’s name.

First try to complete the code on paper.

#include <iostream>

using namespace std;

// A **function prototype** is used to tell the compiler about the

// existence of a function being used in the program (return type,

// name and parameters)

// Declare the function prototype here. It should match the

// function definition.

int main()

{

cout << “Now I will print my Name and My Lab partners name on screen through the function “ << endl;

// A **function call** is a statement that causes a function to

// execute. A function call has to match the function

// definition.

// Make a function call to display message on the screen

cout << “Back to main function after printing names “ << endl;

return 0;

}

// A **function definition** contains the set of operations and

// statements that need to be executed to perform a task.

// A function definition consist of return type, function name,

// parameter list and function body.

// **void function** will not return any value, it will simply execute

// a bunch of statements and then terminate.

void displayMessage()

{

// write an output statement to print your name and your lab // partner’s name

}

1. Write a function for finding average of three numbers. You have to first create the function prototype. Then, write the main program to get the temperature of 3 different zipcodes and find the average of the maximum temperature for these zipcodes by calling the new function. Write pseudo code first.

Try to write your code on paper before start typing the program.

1. A) Type the following program to accept three numbers from the user and print their sum.

First try to complete the code on paper:

#include <iostream>

using namespace std;

// Declare the function prototype here. It should match the

// function definition.

int main()

{

int value1, value2, value3;

// Get three values from the user

cout << “Enter three integers and I will display their sum: “ << endl;

cin >> value1 >> value2 >> value3;

// Make a function call to calculate the sum and display the // sum

return 0;

}

// Function definition to calculate the sum and display it

void showSum(int num1, int num2, int num3)

{

// Complete the code to calculate the sum and display the

// sum on output screen

}

B) Change the above code where instead of displaying the sum in the void function, you will be returning the sum back to the main program, and displaying it.

Hint: You will need to change the return type of the function definition.

1. This program takes two numbers (payRate and hours) and multiplies them to get grosspay. It then calculates net pay by subtracting 15%.
2. Fill in the code (places in bold) and note that the function computePaycheck determines the net pay by subtracting 15% from the gross pay. Both gross and net are returned to the main() function where those values are printed.
3. Compile and run your program with the following data and make sure you get the output shown.

#include <iostream>

#include <iomanip>

using namespace std;

//Function prototypes void printDescription();

void computePaycheck(float, int, float&, float&);

int main()

{

float payRate; float grossPay; float netPay; int hours;

cout << setprecision(2) << fixed;

cout << "Welcome to the Pay Roll Program" << endl; printDescription(); //Call to Description function cout << "Please input the pay per hour" << endl;

cin >> payRate;

cout << endl << "Please input the number of hours worked" << endl;

cin >> hours;

cout << endl << endl;

computePaycheck(payRate,hours,grossPay,netPay);

**// Fill in the code to output grossPay**

cout << "The net pay is $" << netPay << endl;

cout << "We hope you enjoyed this program" << endl;

return 0;

}

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

// printDescription

//

// task: This function prints a program description

// data in: none

// data out: no actual parameter altered

//

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

void printDescription() // The function heading

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

cout << "This program takes two numbers (payRate & hours)" << endl; cout << "and multiplies them to get gross pay " << endl; cout << "it then calculates net pay by subtracting 15%" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

}

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

// computePaycheck

//

// task: This function takes rate and time and multiples them to

// get gross pay and then finds net pay by subtracting 15%.

// data in: pay rate and time in hours worked

// data out: the gross and net pay

//

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

void computePaycheck(float rate, int time, float& gross, float& net)

{

**// Fill in the code to find gross pay and net pay**

1. This program will demonstrate the scope rules.
2. For each comment in bold, place the proper code to do what it says.

NOTE:

area = π *r*2

circumference = 2π*r*

1. Before compiling and running the program, write out what you expect the output to be.

What value for radius will be passed by main (first inner block) to the

findArea function?

What value for radius will be passed by main function (second inner block) to the findCircumference function?

#include <iostream>

#include <iomanip>

using namespace std;

const double PI = 3.14;

const double RATE = 0.25;

void findArea(float, float&);

void findCircumference(float, float&);

int main()

{

cout << fixed << showpoint << setprecision(2);

float radius = 12;

cout <<" Main function outer block" << endl;

cout <<" **LIST THE IDENTIFIERS THAT are active here**" << endl << endl;

{

float area;

cout << "Main function first inner block" << endl;

cout << "**LIST THE IDENTIFIERS THAT are active here**" << endl << endl;

**// Fill in the code to call findArea here**

cout << "The radius = " << radius << endl;

cout << "The area = " << area << endl << endl;

}

{

float radius = 10;

float circumference;

cout << "Main function second inner block" << endl;

cout << "**LIST THE IDENTIFIERS THAT are active here**" << endl << endl;

**// Fill in the code to call findCircumference here**

cout << "The radius = " << radius << endl;

cout << "The circumference = " << circumference << endl << endl;

}

cout << "Main function after all the calls" << endl;

cout << "**LIST THE IDENTIFIERS THAT are active here**" << endl << endl;

return 0;

}

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

// findArea

//

// task: This function finds the area of a circle given its radius

// data in: radius of a circle

// data out: answer (which alters the corresponding actual parameter)

//

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

void findArea(float rad, float& answer)

{

cout << "AREA FUNCTION" << endl << endl;

cout << "**LIST THE IDENTIFIERS THAT are active here**"<< endl << endl;

**// FILL in the code, given that parameter rad contains the radius, that**

**// will find the area to be stored in answer**

}

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

// findCircumference

//

// task: This function finds the circumference of a circle given its radius

// data in: radius of a circle

// data out: distance (which alters the corresponding actual parameter)

//

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

void findCircumference(float length, float& distance)

{

cout << "CIRCUMFERENCE FUNCTION" << endl << endl;

cout << "**LIST THE IDENTIFIERS THAT are active here**" << endl << endl;

**// FILL in the code, given that parameter length contains the radius,**

**// that will find the circumference to be stored in distance**

**}**