**ITCS 1212L**

**Prelab 5**

# Advanced Selection & Functions

**Learning Objectives:**

* **Practice more with if-else-if and switch statements.**
* **To understand the concept of scope**
* **To understand the difference between static, local and global variables**
* **Practice with functions that can return more than a value and their parameter’s scope is not local to the function**
* **Input validation**
* **Practice more with web services**

**These pre-labs are directly related to the lab activities. You don’t need to complete the program. What you need to do is to work on the logic and perhaps some key parts of each program and then complete them in the lab.**

**Prelab activity for Lab 5A:**

Find the difference between two types of returning values.

Implement the following function prototypes (which take hours of work as integer and the rate of pay as float and calculates the net-payment as float) on paper. Also, write a main function which calls both functions. Compare the calculated net values and show the proper message whether or not they match. Name your variables as hours, rate, and netPay.

float calNetPay(int, float);

void calNetPay2(float&, int, float);

#include <iostream>

Using namespace std;

float calNetPay(int, float);

void calNetPay2(float&, int, float);

int main()

{

Int hours(40);

Float rate(10.0), netPay;

netPay = calNetPay(hours, rate);

cout << “first call = “ << netPay << endl << endl;

calNetPay2(netPay, hours, rate);

cout << “second call = “ << netPay << endl << endl;

cout << “as you can see, both functions returned the same answer”;

return 0;

}

Float calNetPay(int hours, float rate)

{

Float netPay;

netPay = hours \* rate;

return netPay;

}

Void calNetPay2(float& netPay, int hours, float rate)

{

netPay = hours \* rate;

}

Also, answer the following questions:

What does the operator & do? It creates a pointer for the variable netPay so we can access it’s memory address and change its value regardless of scope. Create a table and put all of the variables of the program inside the first column. The second column should show the scope of each variable. For example for calNetpay2 function if you called the integer variables hours, its scope will be the function calNetPay2.

Int hours main()

Float rate main()

Float netPay main()

All are called locally in main()

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

Int hours calNetPay()

Float rate calNetPay()

Float netPay calNetPay()

All these are created locally in calNetPay (even though they are named the same as the ones in main)

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

Int hours calNetPay2()

Float rate calNetPay2()

These two are created locally in calNetPay2() despite the same name used in the other functions. Float& netPay is simply a pointer to the variable netPay in main() and is not technically a variable native to the scope of calNetPay2().

**Prelab activity for Lab 5B:**

Create a flowchart showing the following scenario:

1. Present the user with a list of options, 1-3 with proper message as indicated below.
2. Get the user’s choice.
3. If it is one, show the max temperature for zipcode 28223.
4. If it is two, show the min temperature for zipcode 28223.
5. If it is three, exit the program.

In implementing your program, make sure to define menu() function prototype as follows:

int menu();

#include <iostream>

Using namespace std;

int menu();

int main()

{

Int choice;

Choice = menu();

If (choice == 1)

{  
 cout << “The max temp for zip 28223 is 93 degrees!”;

}

Else if (choice == 2)

{

Cout << “The min temp for zip 28223 is 74 degrees!”;

}

Else if (choice == 3)

{

Cout << “You have chosen to exit. Thank you for playing.”;

}

Else

{

Cout << “You have made an invalid selection”;

}

}

Int menu()

{

Int choice;

Cout << “Please make a selection (1-3)” << endl << endl;

Cout << “1. Max temp for zip 28223” << endl;

Cout << “2. Min temp for zip 28223” << endl;

Cout << “3. Exit program” << endl;

Cin >> choice;

Cout << endl << endl;

Return choice;

}

**Prelab activity for Lab 5C:**

You saw the following menu-driven program before which calculates the charges for a health club membership. This time, it is a complete code.

#include <iostream>

using namespace std;

// Function protoypes

char showMenu();

double calculateMembershipRate(char, int);

void displayMembershipRate(double);

int main()

{

// Declare variables

char choice; // To hold menu choice

int months; // To hold the number of months

double membershipRate; // To calculate the membership rates

// Function call to display the menu and return the menu

// choice

choice = showMenu();

// Accept number of months and calculate membership rates if user chooses A, B or C

if(choice != 'D') {

// Get the number of months

cout << "For how many months: " << endl;

cin >> months;

// Make a function call to calculate the membership rates. Send choice, and months as parameters

membershipRate = calculateMembershipRate(choice, months);

// Make a function call to display the final membership rate

displayMembershipRate(membershipRate);

}

else {

cout << "Goodbye !" << endl;

}

return 0;

}

// Function definition to display the menu choice and return the choice to the user

char showMenu()

{

char userChoice;

// Display the menu choices to the user

cout << "\nHealth Club Membership Menu" << endl;

cout << "A. Standard Adult Membership " << endl;

cout << "B. Child Membership " << endl;

cout << "C. Senior Citizen Membership " << endl;

cout << "D. Quit " << endl;

cout << "Enter (A, B, C or D) as your membership choice: " << endl;

cin >> userChoice;

// Return the menu choice back to the user

return userChoice;

}

// Function definition to calculate membership rate for the user based on user choice and number of months

double calculateMembershipRate(char userChoice, int numMonths)

{

// Constants for membership rates

const double ADULT = 40.00;

const double CHILD = 20.00;

const double SENIOR = 30.00;

double rate = 0; // Variable to hold the membership rate

// Switch between user choice

switch(userChoice)

{

case 'A': // calculate the mebership rate for adult

rate = numMonths \* ADULT;

break;

case 'B': // calculate the membership rate for child

rate = numMonths \* CHILD;

break;

case 'C': // calculate the membership rate for senior

rate = numMonths \* SENIOR;

break;

default: // Invalid choice

cout << "Invalid Choice" << endl;

}

// Return the rate back to the main function

return rate;

}

// Function call to display the membership rate to the user

void displayMembershipRate(double finalRate)

{

cout << "The total charges are: $" << finalRate << endl;

}

a) Create the flow chart of this program.

Flow chart

User menu selection

A, B, C D (or anything other than A, B, or C)

User inputs # of months Display “Goodbye !” and end program

Calculates membership cost

Displays membership cost

b) We want to add the following functions to this program. The integer parameter is the number of family members. For each member up to 3 members For the first additional member 15% discount will be assigned, for the second member 10% discount will be assigned , and for the third member only 5% discount will be calculated. Implement the function and call it in the proper place.

double calculateMembershipDiscount(int numberOfMembers);

{

Double discount;

// main() will have to ask for the number of memberships to be passed as the int

If (numberOfMembers == 2)

{

Discount = .85;

}

Else if (numberOfMembers == 3)

{

Discount = .75;

}

Else if (numberOfMembers == 4)

{

Discount = .70;

}

Else

{

Discount = 1;

}

}

// After this function call, main will need to calculate this discount with the finalRate BEFORE

// calling the displayMemberShipRate function.

**Prelab activity for Lab 5D: Winning Division**

Lab5D is about writing a program that determines which of a company’s four divisions (Northeast, Southeast, Northwest, and Southwest) had the greatest sales for a quarter. It should include the following two functions, which are called by your main() function:

* double getSales() is passed the name of a division (string). It asks the user for a division’s quarterly sales figure, validates the input, then returns it. It should be called once for each division.
* void findHighest() is passed the four sales total as parameters. It determines which is the largest and prints the name of the high grossing division, along with its sales figure.

*Input Validation: Do not accept dollar amount less than $0.00*

You need to determine what functions you need for this program and then draw the flow chart for the functions. You need to finish the program and test it in the lab.

Here’s my pseudocode:

Double getSales(string);

Void findHighest(double, double, double, double);

Main()

String division;

Double div1, div2, div3, div4;

Division = “Northeast”;

Div1 = getSales(division);

Division = “Southeast”;

Div2 = getSales(division);

Division = “Northwest”;

Div3 = getSales(division);

Division = “Southwest”;

Div4 = getSales(division);

findHighest(div1, div2, div3, div4);

return 0;

double getSales(string division)

double divSales;

while (divSales < 0)

{

cout << “Please enter the sales for the “ << division << “ division:”;

cin >> divSales;

cout << endl << endl;

if (divSales < 0)

{

Cout << “You entered a value less than 0. Please re-enter the value”;

Cout << endl << endl;

}

}

return divSales;

void findHighest(double div1, double div2, double div3, double div4)

if (div1 > div2 && div1 > div3 && div1 > div 4)

cout << “Northeast wins with total sales of $” << div1 << “!” << endl;

else if (div2 > div3 && div2 > div 4)

cout << “Southeast wins with total sales of $” << div2 << “!” << endl;

else if (div3 > div4)

cout << “Northwest wins with total sales of $” << div3 << “!” << endl;

else

cout << “Southwest wins with total sales of $” << div4 << “!” << endl;

**Prelab for Lab 5E: Safest Driving Area**

Write a program that determines which of the four geographic regions within a major city (North, South, East, and West) had the fewest reported automobile accidents last year. It should have the following two functions, which are called by the main() function:

* int getNumAccidents() is passed the name of a region (string) as parameter. It asks the user for the number of automobile accidents reported in that region during the last year, validates the input, then returns it. It should be called once for every city region.
* void findLowest() is passed the four accident totals as parameter. It determines which is the smallest and prints the name of the region, along with its accident figure.

*Input Validation: Do not accept an accident number that is less than 0.*

Figure out the logic on paper first and show it using flowchart. You need to finish the program and test it in the lab.

Here’s my pseudocode:

Double getAccidents(string);

Void findLowest(int, int, int, int);

Main()

String region;

Double acc1, acc2, acc3, acc4;

region = “Northeast”;

acc1 = getSales(region);

region = “Southeast”;

acc2 = getSales(region);

region = “Northwest”;

acc3 = getSales(region);

region = “Southwest”;

acc4 = getSales(region);

findHighest(acc1, acc2, acc3, acc4);

return 0;

double getSales(string region)

double accNumber;

while (accNumber < 0)

{

cout << “Please enter the number of accidents for the “ << region << “ region:”;

cin >> accNumber;

cout << endl << endl;

if (accNumber < 0)

{

Cout << “You entered a value less than 0. Please re-enter the value”;

Cout << endl << endl;

}

}

return accNumber;

void findLowest(int acc1, int acc2, int acc3, int acc4)

if (acc1 < acc2 && acc1 < acc3 && acc1 < acc4)

cout << “Northeast region had the fewest accidents: “ << acc1 << endl;

else if (acc2 < acc3 && acc2 < div 4)

cout << “Southeast region had the fewest accidents: “ << acc2 << endl;

else if (acc3 < acc4)

cout << “Northwest region had the fewest accidents: ” << acc3 << endl;

else

cout << “Southwest region had the fewest accidents ” << acc4 << endl;

**Prelab activity for Lab 5F: Celsius to Fahrenheit and (vice versa) Temperature Convertor**

You need to prepare a simple flowchart showing the logic of this program and the way each of the functions should be implemented. This lab is asking you to write a program that converts Celsius to Fahrenheit (and vice versa) from a menu option. After selecting which conversion from the menu, the program should accept the Temperature from the user and pass it as a parameter to one of the two functions named f2c\_Convertor or c2f\_Convertor. The function should get and return the temperature after conversion as float.

The formula for converting a temperature for Celsius to Fahrenheit and vice versa are shown below:

F = (9/5) \* C + 32

And

C= (5/9) \* (F – 32)

where:

F is Fahrenheit temperature returned by your function

C is Celsius temperature passed to your function

Int menu();

Float f2c(float);

Float c2f(float);

Main()

Int choice;

Float temp, convTemp;

Choice = menu();

“Enter the temp you wish to convert:”

Cin >> temp;

If (choice == 1)

convTemp = f2c(temp);

else

convTemp = c2f(temp);

endl endl

“The value you entered is = to: “

convTemp

return 0;

float f2c(float temp)

temp = (5/9) \* (temp – 32);

return temp;

float c2f(float temp)

temp = (9/5) \* (temp + 32);

return temp;