**Assignment 3: L3 RAPTOR Self and Peer Assessment**

Write a program that allows the user to input a total dollar amount for an online shopping order and computes and outputs the shipping cost based on the following schedule:

Order Total Ship within USA Ship to Canada

Less than $50.00 $6.00 $8.00

$50.01–$100.00 $9.00 $12.00

$100.01–$150.00 $12.00 $15.00

Over $150.00 Free Free

**Assignment 3: L4 RAPTOR**

.Using RAPTOR (installed from the "Start Here!" page), create a flowchart program to solve the following problem. Attach your .rap file created when you save your flowchart.

Allow the user to enter a series of temperatures in degrees Celsius (C) terminated by the input of −999. For each one, find the corresponding temperature in degrees Fahrenheit (F). The conversion formula is:

F = 9 \* C/5 + 32

**Programming Project 2**

Instructions: The following programming problem can be solved by a program that performs three basic tasks (Input Data, Process Data, Output Results) along with selection and repetition coding techniques. Starting with the program you created in Programming Project 1 and using RAPTOR, design a suitable program to solve this problem.

Problem Statement

A finance company provides loans for motorcycles at different rates depending on how much the total loan amount is and how many payments will be made on the loan. Using the information in the table below, write a program that will calculate the monthly payment based on user inputs of loan amount and number of monthly payments. The user will NOT input the percentage rate, as this will be determined by the program code based on user input of loan amount and number of payments. The output will display the loan amount, number of payments, monthly payments and the interest rate of the loan. Use a loop to allow users to enter as many sets of data as desired. At the end of each loop, ask the user if he or she would like to Exit the program (Y for Yes or N for No). If yes, clear the variables and repeat the input, processing and output loop. If no, exit the program.

Problem Statement

Amount of Loan # of Payments Interest Rate Applied

$500 - $ 2,500 6-12 8%

13-36 10%

37-48 12%

$2,501 - $10,000 6-12 7%

13-36 8%

37-48 6%

$10,001 or above 6-12 5%

13-36 6%

37-48 7%

If the user enters data that is "out of bounds" (loan amount/number of payments below or above minimum/maximum in table), display an error message explaining the situation to the user and ask for the loan amount or number of payments (whichever one was out of bounds) again. Message Example: "We do not finance loans below $500."

You MUST use Modular Programming techniques by using Sub Modules (Sub Charts in RAPTOR) in your program. Your "main" module should not be very large.

Begin with the program you created in Programming Project 1. Modify the program so that it completes the additional tasks.

**Programming Project 3**

.Instructions: The following programming problem can be solved by a program that uses three basic tasks-Input Data, Process Data, and Output Results. To process the data, it uses loops, arrays, decisions, accumulating, counting, searching and sorting techniques. Use RAPTOR to design a suitable program to solve this problem.

Problem Statement

Assume the Scores array is parallel to the Players array (both arrays are below).

Scores array

Scores[0] = 198

Scores[1] = 486

Scores[2] = 651

Scores[3] = 185

Scores[4] = 216

Scores[5] = 912

Scores[6] = 173

Scores[7] = 319

Scores[8] = 846

Scores[9] = 989

Players Array

Players[0] = "Joe"

Players[1] = "Ann"

Players[2] = "Marty"

Players[3] = "Tim"

Players[4] = "Rosy"

Players[5] = "Jane"

Players[6] = "Bob"

Players[7] = "Lily"

Players[8] = "Granny"

Players[9] = "Liz"

Write a looping program that presents the user with 3 options:

1) Sort Output by Players

2) Sort Output by Scores

3) Exit Program

When the first option is selected, sort the Players array in alphabetical order, keeping the Scores array parallel. Add code that determines the highest and lowest scores in the list. Include code to display each player’s score and name in the sorted order. Below the sorted list display the highest and lowest scores in the list and the name of the player who received that score. Your sort by Player output display should look like this:

Scores Sorted by Player:

486 Ann

173 Bob

846 Granny

912 Jane

198 Joe

319 Lily

989 Liz

651 Marty

216 Rosy

185 Tim

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

989 Highest Score by Liz

173 Lowest Score by Bob

When the second option is selected, sort the Scores array in numerical order, keeping the Players array parallel. Add code that determines the average score of the entire list. Include code to display each player’s score and name in the sorted order. Below the sorted list display the average of all scores in the list. Your sort by Scores output display should look like this:

Players Sorted by Scores:

173 Bob

185 Tim

198 Joe

216 Rosy

319 Lily

486 Ann

651 Marty

846 Granny

912 Jane

989 Liz

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

498 Average Score

You may use either the Bubble Sort or the Selection Sort algorithms.

You MUST use Modular Programming techniques by using Sub Modules (Sub Charts in RAPTOR) in your program. Your "main" module should not be very large.