EECE104P: Object Oriented Programming Using C++

Tutorial Sheet 4: Loops and Decision Making (continued)

This tutorial sheet contains some programming exercises related to loops (for, while, do while), and also decision making based on if, else if, else, and switch constructs. You are expected to complete them during the lab session. Save the corresponding C++ files

1. Write a C++ program, which will sum the following infinite series to the desired level of convergence (n varying from zero to infinity):

$$S = \sum (-1)^n x^{n/2} / n(n+1)$$

2. Write a C++ program, which will print out the values of the function

$$F(x) = (x^2 + 2x + 3) / (x-20)$$

For N equally spaced values in the range $x_{min} < x < x_{max}$, where x_{min}, x_{max}, and N are user defined.

3. Write a C++ program which will take a decimal integer as an input, and print out its binary equivalent.

4. Write a C++ program which will find and printout all the prime numbers n_p in the range $n_{min} < n_p < n_{max}$, where n_{min} and n_{max} are user defined.

5. Write a C++ program which will ask the user to give a positive integer (say n), and then printout the values of all the factorials: 1!, 2!, 3!,..., n!. That is if the user gives n = 3, then the values of 1!, 2!, and 3! are printed out. Don't compute each factorial from scratch, rather use the recursion relation n! = n(n-1)! to generate them within the for loop.

EECE104P: Object Oriented Programming Using C++

Tutorial Sheet 5: Structures and Enumerations

This tutorial sheet contains programming exercises related to user defined data types structures and enumerations. Save the corresponding C++ files.

1. A point in three-dimensional space can be expressed by three numbers (x; y; z), which are its x-, y- and z- coordinates, respectively. A sum of two points (x1; y1; z1) and (x2; y2; z2) is a new point (x1 + x2; y1 + y2; z1 + z2). Write a C++ program which defines a structure called point to model a point. Define three points say p1, p2, and p3, and take the values of p1 and p2 as input from the user. Then set p3 as the sum of coordinates of p1 and p2, and print out the value of p3.

2. Define a structure called Distance which contains three variables meters, centimeters, and millimeters. Keep meter as a float variable, while centimeters and millimetres should be of the type int. Define three variables d1, d2, and d3, of type Distance, and take the values of d1 and d2 as input from the user. Then assign to d3 the value obtained by summing d1 and d2, print out its value. While performing these sums, use the fact that 1 cm = 10 mm, and 1 m = 100 cm.

3. Define a structure called Room, whose length, breadth and height are of data type Distance as defined in the previous problem. Read from the user the three dimensions of the room, and print out the volume of the room in cubic meters.

4. Create a structure called employee that contains two members: employee number (type int) and employee salary in INR (type float). Ask the user to input the data for three employees and then print it out.

5. You may remember that in Tutorial Sheet 2 we wrote a program to add two fractional numbers of the form a/b and c/d. Write a C++ program which defines a structure called Fraction whose two members represent the numerator and the denominator of a fractional number, and both are of type int. Define three variables f1, f2, and f3 of the type Fraction, take f1 and f2 as input from the user, and then obtain f3 as the sum of f1 and f2, using the rules of addition of a fractions. Finally, the program should print out the value of f3.

6. Define an enumeration called Vowels with members A, E, I, O, and U. Then define two variables v1 and v2 which are of the type Vowels, and initialize them to I and A, respectively. Print out whether I comes before A, or the other way round. Look at example on pages 148-149 of Lafore's book to understand the enum data type.

EECE104P: Object Oriented Programming Using C++

Tutorial Sheet 6: Functions

This tutorial sheet contains programming exercises related to the use of functions. Save the corresponding C++ files.

1. Write a C++ program which defines a float function called Volume, and uses it to calculate the volume of a rectangular cuboid. The program should ask the user for the height, length and breadth of the cuboid, and then call the function Volume to calculate its volume and print it out. Thus, function Volume should have three float arguments corresponding to height, length and breadth, and should return the value of the volume of the cuboid.

2. Write a program that will use a function called power to compute the n-th power of a floating point number x, by multiplying it with itself n times $x^n = x * x * x * \ldots\ldots x$. The function should take x and n as arguments, and return the value of x^n.

3. Write a program which will swap (interchange) two integers n1 and n2 by using a function called swap, and then the swapped values are printed out. Note that in this case you should use arguments by reference.

4. Write a C++ program which uses a void function to calculate the volume and total surface area of a cone, from its radius (r) and slant height (l). Note that in terms of these quantities the volume of the cone is $V = 1/3\pi r^2 h$ (h is the height of the cone), and surface area is $S = \pi r^2 + \pi r l$, with the relation between h and l being $l^2 = h^2 + r^2$. Because this function is void, so pass the values of V and S to the calling program using arguments by reference. Thus, this function should have four float arguments: r and l regular arguments, and V and S reference arguments. Also calculate the value of π inside the function using the relation $\pi = 4 \tan^{-1}(1.0)$. The library function for $\tan^{-1}(x)$ in C++ is atan(x), and to use it you have to include the header file math.h.

5. Write a C++ program to carry out addition, subtraction, multiplication, and division on fractional numbers of the form a/b etc. To define a fractional number define a struct fraction. Then define four functions fadd(), fsub(), fmul(), and fdiv() to carry out these mathematical operations. These functions should take structure fraction as arguments, and also should return the value of the same type, which is the result of the operation. Take two fractional numbers as input in the main program, and then print the results of the four mathematical operations on them computed by using these functions.

EECE104P: Object Oriented Programming Using C++

Tutorial Sheet 7: Functions and Strings

This tutorial sheet contains programming exercises related to the use of functions and strings. Save the corresponding C++ files.

1. Write a C++ program to create a database of students with the following information for each student:

ID (integer)

Marks in 5 subjects for 3 exams (Midsem, Lab, Endsem with a weightage of 25, 25, 50 respectively).

Create a function to compute total marks for a subject given the marks in the three exams, and create another function to compute the overall result (percentage) for a student. Ask the user for an ID, compute the result for that student and print it back to the user.

2. After completing the above exercise, repeat the same by adding the name field in the database for students.

3. Write a C++ program, which takes a string as an input from the user, and prints the reverse of that string. If the string is same as its reverse, print: "It is a palindrome".

4. In Fibonacci series, every number is the sum of the two preceding numbers. Write a program to take two numbers x1 and x2, and a length N, as input from the user. Using the two numbers, create a function to generate next N numbers for the Fibonacci series. Save the entire series in an array. Now, print the entire Fibonacci series inside main(), not inside the function used for creating the series. *(Hint: pass the pointer to the array as an input to the function).*