CSL303 IOOM

Programming Format Guidelines

Writing code:

- For every class write constructors (default as well as parameterized), copy constructor, getter & setter methods and destructor.
- Write driver program to demonstrate the working of the code wherever required.

Grading criteria

- *Does it work?* Your programs should compile without warnings or errors. They should give the correct answer, both on the example inputs given in the assignment (if provided) and possibly on arbitrary new examples we may create during testing. You can get at most 30% of the maximum score if the program does not run.
- Is it found correct also after reading the code? (The test set may not have revealed all bugs.)
- The quality of the solution: e.g., is it unduly complex?
- Comments and structuring.

Here are some details of the last point.

Comments

- A program header containing the name of the name, enrollment, assignment number and date, and a very short description (shorter than the external documentation) of what the program is about.
- Brief comments before the definition of each major procedure or group of procedures detailing:
  - What does it do?
  - The use of each parameter.
  - Preconditions that must be met when the procedure is called.
  - Output, return values and side effects (if any).
- Similar explanation is expected for each non-obvious variable, structure, or class declaration.
- Brief comments at any point in the program at which the reader needs help in understanding. Comments which merely rephrase code are unnecessary, distracting, and a potential maintenance hazard (often code is changed, but the comments are not). Comments should edify the code, not repeat it. Use of good variable names and function names can help keep code self-commenting.

Structuring

- Make the program largely self-explanatory by arranging it in an intelligent way and giving informative names to your variables and procedures. Then fewer comments are needed since you must only comment the non-obvious.
- Maintain consistent indentation as it helps to discover many syntactic errors early.

Naming conventions

- Names of variables, constants and functions must clearly indicate the function and type of the named object.
- Follow the convention that constants should be all capital letters, with multiple words separated by underscore "_".
- Within variable and function names, multiple words can be separated by either capitalizing or an underscore.
- The names must be self-explanatory not only for you but also for others reading your code.

Global variables

Try to avoid global variables using methods for structuring programs taught in the course.

CSL303- Introduction to Object Oriented Methodology

Lab Assignment – 2 Batch 1

Submission Date: Monday 23 November, 2020 Time 11.50 PM

Marks : 20

Q.1) Overload 'new' and 'delete' operators to manipulate objects of the class Student. The Student class has data members such as 1) char *name, 2) int roll_no, 3) char *branch etc. The overloaded 'new' and 'delete' operators must allocate memory for the Student class object and its data members. Create a database for 5 students and store in an array. Overload the 'subscript' operator for bounds checking while accessing i^{th} student object. [6 M]

Q.2) Design a class Polar which describes a point in the plane using polar coordinates radius 'r' and angle 'a'. Overload the '+' operator to add two polar objects. [4 M]

Note:- We can't add two polar objects directly. This requires first the conversion of points into rectangular coordinates, and then adding the corresponding rectangular coordinates and finally converting the result back into polar coordinates. Following are the conversion formulas. Include the required library files in your program.

1) x= r * cos(a);

2) y= r * sin(a);

3) a= atan(y / x); // arc tangent

4) r= sqrt(x*x+y*y);

Q. 3) Write a program for creating a doubly linked list of template type. The program should allow ordered insertion and deletion of a given node. User should be able to create doubly linked list of decimal numbers and strings. Program should also handle exceptions such as exception that may arise while allocating memory for a node of linked list, exception if user tries to delete a node that does not exist and null pointer exception.

[7 M]

Q.4) Write a program which divides two complex numbers by overloading the divide operator. Handle the error such as division-by-zero using exceptions. [3 M]

CSL303- Introduction to Object Oriented Methodology

Lab Assignment – 2 Batch 2

Submission Date: Monday 23 November, 2020 Time 11.50 PM

Marks : 20

1. Define a class ArrayNumber (duplicate values permitted) of 20 integers. The values of the integer are between 1 and 10. Overload the subscript operator [].Overload 'new' and 'delete' operators to manipulate objects of the class ArrayNumber. [2 M]

2. Define a class Vector which can hold 15 elements of type int. Write a member function read () which reads the values from the input stream and assigns them to the Vector object. Overload the operator plus, minus, multiplication and division for the class Vector above which adds two vectors. Define the above mention operator as a friend function. The result must be returned as an object of the class Vector.
[3M]

3. Write a program to overload << and >> operator to I/O the object of a class. [2 M]

4. Define a DateofBirth (DOB) class with
   – An overloaded increment\decrement operator to change DateofBirth
   – An overloaded **+=** operator
   – A function to test whether birth year is leap years or not.
   – A function to determine if a day is last day of a month [3 M]

5. Implement exception handling for handling both when the Array is empty and when the Array is full. And class Array which can hold 15 element of type double. Define a member function int Get (int index); which returns the indexth element of the array if index in between 0 to 14 and throws an exception if index is out of bounds. Catch the exception in the main program and print an error.
[5]

6. Implement operation on array by using template. Write three function templates for
   a. Sorting a list of arrays
   b. Find out cube of a given array element.
   c. Binary search that can be used to search an array of elements of any type and returns the index of the element, if found. Give both the function prototype and the function definition for the template. Assume a class template Array <T> is available.
[5]

CSL303- Introduction to Object Oriented Methodology

Lab Assignment – 2 Batch 3

Submission Date: Monday 23 November, 2020 Time 11.50 PM

Marks : 20

Que. 1) Array Class

Pointer-based arrays have a number of problems. For example, a program can easily "walk off" either end of an array, because C++ does not check whether subscripts fall outside the range of an array. Arrays of size *n* must number their elements 0, ..., *n* – 1; alternate subscript ranges are not allowed. An entire non-char array cannot be input or output at once, each array element must be read or written individually. Two arrays cannot be meaningfully compared with equality operators or relational operators (because the array names are simply pointers to where the arrays begin in memory and, of course, two arrays will always be at different memory locations). One array cannot be assigned to another with the assignment operator(s) (because array names are const pointers and a constant pointer cannot be used on the left side of an assignment operator). These and other capabilities certainly seem like "naturals" for dealing with arrays, but pointer-based arrays do not provide such capabilities. However, C++ does provide the means to implement such array capabilities through the use of classes and operator overloading.

[8]

Create a powerful **Array class** that can provide following capabilities:

Array class must have data member array of size *n*.

Overload subscript operator for range checking to ensure that subscripts remain within the bounds of the Array.

Overload assignment operator: The class allows one array object to be assigned to another with the assignment operator.

Overload stream extraction/stream insertion: Entire Arrays can be input or output with the stream extraction and stream insertion operators, respectively.

Overload equal comparison(==) operator: Array comparisons can be made with the equality operator.

Overload not equal comparison (!=) operator: Array comparisons can be made with the not equality operator.

Que. 2) Give C++ code for reading a file, counting the frequency of keywords (No. of keywords may be between 2 – 8 words), and storing the frequency of each individual in a file. Write appropriate exception for:

a) IOException
b) EOFException
c) FileNotFoundException

[6]

Que. 3) Implement binary search tree operations insert, delete and search for different data values (int, float, char) with the help of template. [6]

CSL303- Introduction to Object Oriented Methodology

Lab Assignment – 2 Batch 4

Submission Date: Monday 23 November, 2020 Time 11.50 PM

Marks : 20

Q.1) Write a menu driven operator overloading program for manipulating matrices. Overload the operators such as **>>, <<, +, -, *, = =** using friend function. Input the matrices from user. Overload 'new' and 'delete' operators to manipulate objects of the same class. [6 M]

Q.2) Write a program to overload the unary operators '+ +' and '- -' for processing counters. It should support both upward and downward counting. It should also overload the '+' operator for adding two counters and '=' operator for storing the result in another counter. [4 M]

Q.3) Write a class template to represent a generic vector. Use dynamic data members instead of arrays for storing vector elements. Perform following operations for vectors. [7 M]

a) Create Vector:- While creating vector, catch the exception raised by new operator and take appropriate action.

b) Add two vectors:- While adding vectors if two vectors are of different size then raise the exception and handle it. If elements of two vectors are of different data type then 1) if the data types are convertible then convert and add vector elements otherwise, 2) raise the exception and handle it.

c) Multiply vector by a scalar value

d) Display the contents of a vector

Q.4) Write a function template to perform linear search in an array. [3 M]