Object oriented programming projects

All projects mentioned below must meet the following guidelines and requirements:

1. Project must be made individually or in grpups of maximum two students.
2. In the project it is necessary to create at least one class with a constructor and appropriate methods. It is also necessary to use at least one of the STL containers e. g. vector, map or other.
3. If it makes any sense for the particular project, the program should also make use of class inheritance, interfaces, exception handling, virtual functions and recurrence.
4. There should be at least minimal input values validation. For instance, when the program expects that a user enters a digit, the program should check if the provided string is actually a digit or not and let the user know when the wrong input was provided. It is unacceptable that the project terminates or falls into an infinite loop when the wrong input is provided.
5. There is no need to create the graphical user interface. It is enough that the project runs in the terminal and text mode.
6. In the main menu, there should be an option that terminates the program. It should also be possible to return to the main menu from any submenu.
7. The project can use external open source libraries, like for instance database server sqlite etc.
8. All changes made during the program execution should be saved (e. g. in files) and restored upon the program startup.
9. The project should be compiled with warnings set to level three. The source code should be written in such a way that there should be no warnings during the compilation. Only a few warnings are acceptable provided that they are difficult to remove.
10. The source code should be well formatted and cleaned of all unused code. There should be comments in the source code explaining the usage of all functions and classes. Most variables (except for loop indexes) should have the names that are meaningful.

Projects topics

The maximum grade for the project is given in parenthesis.

1. Design and implement classes for a simple calculator. (3.0)
2. Write a number guessing game. The program is going to randomly select an integer from 1 to 100. You will keep guessing numbers until you find the program's number, and the game will tell you each time if your guess was too high or too low. (3.0)
3. Design and create the Date class which makes it possible to perform the standard operations on dates (e.g. difference between two dates, adding days to the date, printing the date in different formats). Write a program that demonstrates the use of the class (3.0)
4. Design and implement classes for a noughts and crosses game (3.0)
5. Design and implement classes for a simple clock and calendar application. (3.0)
6. Write a program that will read from a file two or more words and will say if they all are anagrams. Anagrams are two or more words composed of the same characters, but in a different order. (3.0)
7. Design and implement classes for a system of deck of cards. Create a method shuffle to make sure cards are not in exact order. (3.0)
8. Write a program that will read from a file two or more words and will say if they are ordered words. An *ordered word* is a word in which the letters appear in alphabetical order. (3.0)
9. Change Return Program - The user enters a cost and then the amount of money given. The program will figure out the change and the number of quarters, dimes, nickels, pennies needed for the change. (3.0)
10. Unit Converter (temp, currency, volume, mass and more) - Program that converts various units between one another. The user enters the type of unit being entered, the type of unit they want to convert to and then the value. The program will then make the conversion. (3.0)
11. Credit Card Validator - Program that takes in a credit card number from a common credit card vendor (Visa, MasterCard, American Express, Discoverer) and validates it to make sure that it is a valid number (look into how credit cards use a checksum). (3.0)
12. Write a simple ToDo List application. (3.5)

13. Design and implement classes for the program that is used to transcode text files. For instance, the program should transcode text files from utf-8 to unicode and vice versa. (3.5)

14. Currency Converter (3.5)

15. Design and implement classes for the Hangman game. In the game of Hangman, the computer chooses a word at random from a given list of words. This word is the answer. The player then tries to guess the word, by guessing one letter at a time. Whenever the user guesses a letter that is in the answer, all occurrences of that letter are revealed to the user. The game ends when the user has guessed every letter in the word, before he reaches the allowed number of strikes. (3.5)

16. Design and implement classes for casino number guessing games. This is a simple text base number guessing game in which a player can deposit his money to play. From this amount he can bet on numbers between 1 to 10. If he wins he gets 10 times the money otherwise lost his money. (3.5)

17. Write a simple yellow cards (notes) application. (3.5)

18. Write a battleship game. (3.5)

19. Contact Management System (3.5)

20. Design and implement classes for a simple calendar application. (3.5)

21. Write a program that will replace every occurring instance of a piece of text in a group of text files with another one. (3.5)

22. Write a program that will read a .cpp and .h file and strip all block comments from program text. The resulting text should be saved in another file. (3.5)

23. Write a program that will read some .cpp files and then print out text statistics like for instance: number of characters, number of words, number of lines, percentage of comments etc. (3.5)

24. Mortgage Calculator - Calculate the monthly payments of a fixed term mortgage over given n-th terms at a given interest rate. Also figure out how long it will take the user to pay back the loan. Add an option for users to select the compounding interval (Monthly, Weekly, Daily, Continually). (3.5)

25. Design and implement classes for a Snake Ladder game. (3.5)

26. Memory Puzzle - A board full of overturned cards. There is a pair for each card. The player flips over two cards. If they match, then they stay overturned. Otherwise they flip back. The player needs to overturn all the cards in the fewest moves to win. (4.0)

27. Sliding Puzzle - A 4x4 board of numbered tiles has one missing space and is randomly set up. To win the game, the player must slide tiles over to put the tiles back in order. (4.0)

28. Nibbles - A worm or snake constantly moves around the board. The player controls the direction the "head" of the worm moves, and the worm must try to eat apples that randomly appear. Eating an apple causes the worm to grow in length. The game ends if the worm crashes into the edge of the board or into itself. (4.0)

29. Sudoku Game (4.0)

30. Connect Four - two players of different colors drop their tokens on an upright board. The player who makes four tokens in a row, column, or diagonal wins. (4.0)

31. Product Inventory Project - a program that manages an inventory of products. Create a product class which has a price, id, and quantity on hand. Then create an inventory class which keeps track of various products and can sum up the inventory value. (4.0)

32. Design and implement classes for an xml parser. (4.0)

33. Design and implement classes for a Json parser. (4.0)

34. Design and implement classes for a HTML parser. (4.0)

35. Design and implement classes for a password manager. (4.0)

36. Design and implement classes for a processor monitoring application. (4.0)

37. Design and implement classes for a program that transcodes image files using an external image processing library. (4.0)

38. Design and implement classes for an political election system (4.0)

39. Design and implement classes for a question bank. (4.0)

40. Recipe Creator and Manager - create a recipe class with ingredients and put them in a recipe manager program that organizes them into categories like deserts, main courses or by ingredients like chicken, beef, soups, pies etc. (4.0)

41. Flower Shop Ordering To Go - create a flower shop application which deals in flower objects and use those flower objects in a bouquet object which can then be sold. Keep track of the number of objects and when you may need to order more. (4.0)

42. Family Tree Creator - create a class called Person which will have a name, when they were born and when (and if) they died. Allow the user to create these Person classes and put them into a family tree structure. Print out the tree to the screen. (4.0)

43. Quiz Maker - make an application which takes various questions from a file, picked randomly, and puts together a quiz for students. Each quiz can be different and then reads a key to grade the quizzes. (4.0)

44. CSV File Utility - a program that reads a file of records, sorts them, and then writes them back to the file. Allow the user to choose various sort styles and sorting based on a particular field. (4.0)

45. Design and implement classes for a text editor including its UI. (4.0)

46. Design and implement classes for a vending machine. Use case list: select item and get price, accept bills/coins, dispense items purchased and return change, refund when cancelling the request. Possible exceptions: Sold out, Not fully paid, Not enough changes. (4.0)

47. Design and implement classes for an elevator system. (4.0)

48. Design and implement classes for a Traffic Light System. Consider the scenario where traffic is coming from all 4 Directions (N,S,E,W) and pedestrians. (4.0)

49. Design and implement classes for student report cards. The system makes it possible to add, modify and display details of a student's marks. The system also provides a function to calculate grades based on marks obtained by students. (4.0)

50. Design and implement classes for the supermarket billing system. A customer can purchase a product and his invoice is generated. Administrators can create, modify, view and delete product records. (4.0)

51. Write a converter from plain text file to HTML file. The plain text has no formatting information however it may have centered headlines, numbered sections, paragraphs, lists, and URIs. (4.0)

52. Electricity Billing System - the project aims at serving the department of electricity by computerizing the billing system. It mainly focuses on the calculation of units consumed during the specified time and the money to be paid to electricity offices. This computerized system will make the overall billing system easy, accessible, comfortable and effective for consumers. (4.0)

53. Othello - on a grid, a black and white player places tiles of their color on the board. The opponent's tiles between the newly placed tile and that player's existing tiles are flipped to become the color of the player's tiles. The game ends when the board fills up and the player with the most tiles of their color wins. (4.0)

54. Design and implement classes for a command line interpreter. (4.5)

55. Design and implement classes for an internet chat. (4.5)

56. Design and implement classes for a students database. (4.5)

57. Design and implement classes for a flight reservation system (4.5)

58. Design and implement classes for a chess game. The program should make it possible to make moves, detect incorrect moves and detect the checkmate. It should also be possible to undo the previous movement. (4.5)

59. Design and implement classes for a tetris game. (4.5)

60. Attendance Management - This project manages attendance of students. Admin can register students and create username and password for students. Students can login, mark attendance on the same day & also can see the history of attendance. (4.5)

61. Patient and Doctor Scheduler - create a patient class and a doctor class. Have a doctor that can handle multiple patients and set up a scheduling program where a doctor can only handle 16 patients during an 8 hr work day. (4.5)

62. Design and implement classes for a task management system. (4.5)

63. Design and implement classes for representing linear algebra. (4.5)

64. Design and implement classes for a parking lot. (4.5)

65. Student Grade Book Application – keep track of students (with a student class that has their name, average, and scores) in a class and their grades. Assign their scores on tests and assignments to the students and figure out their average and grade for the class. (4.5)

66. Create a library of functions and classes for manipulation of matrices of complex numbers. (4.5)

67. Flood It - a grid of six colors of tiles starts off randomly. The player can do a "flood fill" on the top left tile, changing the color of any adjacent tiles of the same color. The player wins if they are able to make the entire board a single color within a certain number of moves. (4.0)

68. Design and implement classes for the banking system. The system should provide as a minimum the following functionality: a customer can deposit, withdraw and transfer an amount in his account. Administrators can create, modify and delete accounts. (4.5)

69. Design and implement classes for a DVD rental. (5.0)

70. Design and implement classes for a car rental. (5.0)

71. Design and implement classes for a program that manages public transport schedules. (4.5)

72. Design and implement classes for auction software. (5.0)

73. Design and implement classes for a book reader system which provides the following functionality: searching the database of books and reading a book, user membership creation and extension, only one active user at a time and only one active book by this user. (5.0)

74. Design and implement classes for a hotel booking system where a user can search a hotel in a given city and book it. (5.0)

75. Design and implement classes for a dating system. The system should be able to find matches based on various criteria. (5.0)

76. Airline Reservation System - create a reservation system which books flights as well as airplane seats. It charges various rates for particular sections of the plane. (5.0)

77. Design and implement classes for an event booking system. (5.0)

78. Design and implement classes for an air traffic control system. (5.0)

79. Design and implement classes for a cinema booking system, users can select movie, room, time, seats and place order. The system includes following object: movie, room, track (in which room when which movie is showing), seat, user, order(a user books which track and which seats) Additionally, this system can be extended to be used for cinema administrator to plan movie(when to plan which movie in which room) in order to get the most audience and income. (5.0)

80. Design and implement classes for the library management system. A student can issue a book and deposit it within 15 days. Students are allowed to issue only one book. Administrators can add, modify or delete records. Create a book class with a title, page count, ISBN and whether or not it is checked out or not. Manage a collection of various books and allow the user to check out books or return books. Generate a report of those books overdue and any fees. Also allow users to put books on reserve. (5.0)

81. Write a program that will work as a simple spreadsheet application. The user should be able to load, save the spreadsheet and perform the most common operations on the spreadsheet. (5.0)

82. Write a sports betting system. (5.0)

83. Design and implement classes for the municipal transport system. The program should make it possible to add new bus/tram stops, search for connections, add time schedules for each stop etc. (5.0)

84. Bug Tracking System - the main aim of the project is to manage the errors or bugs occurring during software development phase and cycle. After the implementation of this project, the employees can update the issue details, solve issues and update the system from any location with internet access. (4.5)

85. Event Management System - the project's main objective is to control or manage the activities and duties to be performed by various event conductors such as attendees, organizers, event reviewers, and authors. (5.0)

86. Stock Management System - the main objective of this project is to manage stock for a company or organization, and take care of sales and purchase of products. This

project includes various modules and features to add, edit, view and delete stock-management-related things in the system database. (5.0)

87. Travel Agency Management System (5.0)

88. Railway Reservation System (5.0)