| ***Computer Engineering Department*** |
| --- |
| ***CE100L: Computing Fundamentals & Programming*** |

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 17/01/2022*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 13A. Challenging Problem Solving in C++**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
| NIMRA MAQBOOL | BSCE21012 |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The objective of this lab is to understand open ended problem solving.

## **Equipment and Component**

| **Component Description** | **Value** | **Quantity** |
| --- | --- | --- |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A challenge is something new and difficult which requires great effort and determination.Every problem has a solution, we just need to think and try to solve it. We can solve every problem; we just need some courage to solve it. It depends on the problem; how much courage we require to solve that problem.



Figure 1: \*What is Open Ended Problem Solving??

\*https://medium.com/@aitisam.zafeer88/problem-solving-challenge-322a3ee7db23

**Lab Task**

1. Suppose you own a hospital, and you have 30 doctors appointed in the hospital. You need to maintain the record of all the registered doctors and display their data when required. Create a structure Doctor, containing registration number (int), name (char []), address(nested), date of joining(nested) as structure members. Write a program that accomplish the following tasks

* Main function should ask the user to enter id and password to proceed further.
* Main function should display a menu either to enter the record or display the record. Program should call the function according to the user's choice.
* If the user chooses to enter the record, the program should store the data in a structure array of 30 members. Program should ask the user if he wants to enter the next record.
* If the user chooses to display the record and there is a record present, the program should display the list of records (in ascending order of date of joining), if there is no record present then it should display a message of “no record found”.
* Record displayed by point number 4 should be sorted in descending order of date of joining of doctors by using a sort function

|  |
| --- |

2. You are supposed to make a C++ Application for the Arfa CAFE management system. Suppose there are two employees who manage the counter, their employment IDs are 431 and 532. The program first login the employee with their ID. Do make a switching mechanism for the employee with a special flag variable. There they can make a transaction of only 2 out of 10 items for a student containing the items based on the below list.

Serial No.

item

Name

Price

1 Tea 30

2 Small Pizza 40

3 patties 40

4 Pastry 40

5 Samosa 10

6 Green Salad 30

7 Sandwich 70

8 Burger 100

9 Fries 100

10 Coffee 50

Display the list before purchase. Your system should serve n number of students, after purchasing anything the system display menu again. The program should display the Name of CAFE at the top of the console. When the employee switches the system to another employee, the program should display the total number of transactions he has done with the system and the program starts again to login system.

| void totalBill() {  int serialNo;  int price[9];  price[0]=0;  price[1]=0;  price[2]=0;  price[3]=0;  price[4]=0;  price[5]=0;  price[6]=0;  price[7]=0;  price[8]=0;  int price0;  int price1;  int bill = 0;  do {  cout << endl << " CHOOSE ANY OF THE serial No. GIVEN BELOW :" << endl;  cout << "1.for tea and 2 small pizza " << endl;  cout << "2.patties" << endl;  cout << "3.pastry" << endl;  cout << "4.samosa" << endl;  cout << "5.green salad" << endl;  cout << "6.sandwich" << endl;  cout << "7.Burger" << endl;  cout << "8.Fries" << endl;  cout << "9.coffee" << endl;  cout << "10. EXIT " << endl;  cin >> serialNo;  switch (serialNo) {  case 1: {  cout << "please enter price of Tea = ";  cin >> price0;  cout << "please enter price of 2 small pizzas = ";  cin >> price1;  price[0] = price0 + price1;  cout << "price is:" << price[0] << endl;  break;  }  case 2: {  cout << "please enter price of patties = ";  cin >> price[1];  cout << price[1] << endl;  break;  }  case 3: {  cout << "please enter price of pastry = ";  cin >> price[2];  cout << price[2];  break;  }  case 4: {  cout << "please enter price of samosa = ";  cin >> price[3];  cout << price[3];  break;  }  case 5: {  cout << "please enter price of green salad = ";  cin >> price[4];  cout << price[4];  break;  }  case 6: {  cout << "please enter price of sandwich = ";  cin >> price[5];  cout << price[5];  break;  }  case 7: {  cout << "please enter price of burger = ";  cin >> price[6];  cout << price[6];  break;  }  case 8: {  cout << "please enter price of fries = ";  cin >> price[7];  cout << price[7];  break;  }  case 9: {  cout << "please enter price of coffee = ";  cin >> price[8];  cout << price[8];  break;  }  case 10: {  cout << "you have choose to exit.";  exit(1);  break;  }  default: {  cout << "you have enter invalid number.";  break;  }  }  } while (serialNo >= 1 && serialNo <= 10);  bill = bill + price[0] + price[1] + price[2] + price[3] + price[4] + price[5] + price[6] + price[7] + price[8];  cout << "bill is" << bill << endl;  } |
| --- |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

| Performance metric | Mapping (task no. and description) | | Max marks | Exceeds expectation | Meets expectation | Does not meet expectation | Obtained marks |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Realization of experiment (a) | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 2 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | Code commenting | 5 | Observes lab safety rules; handles the equipment and parts with care and adheres to the lab disciplinary guidelines aptly (4-5) | Generally observes safety rules and disciplinary guidelines with minor lapses (2-3) | Disregards lab safety and disciplinary rules (0-1) |  |
| 5. Data collection (c) | 1 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | Documentation | 5 | Timely documented (4-5) | Late documented (2-3) | Not documented (0-1) |  |
|  | Max Marks (total): | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_