|  |
| --- |
| **COMPUTER Engineering Department - ITU** |
| **CE101L: Object Oriented Programming Lab** |

|  |  |
| --- | --- |
| **Course Instructor: Usama Bin Shakeel** | **Dated: 24/05/2022** |
| **Teaching Assistant: Aqsa Khalid** | **Semester: Spring 2022** |
| **Lab Engineer: Nadir Abbas** | **Batch: BSCE2021** |

# **Lab 11A. Problem Based Learning by use of Objects and Classes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 observe the basic knowledge of programming classes in C++.

## **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**

**Open-ended problem** is a problem that has several or many correct answers, and several ways to the correct answer(s). The Open-Ended Approach provides students with "experience in finding something new in the process"(Shimada 1997). It is basically facilitating the development of creative problem solving skills.

Diagram

Description automatically generated

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

**Lab Task**

**Task A: [Marks: 40]**

Write a program in which a student is an object in a university management System. Analyze the concept and identify the data members that a Student class should have. Also analyze the behavior of student in a university management System and identify the methods that should be included in Student class.

|  |
| --- |
| Function.h:  #include <iostream> #include <string> #include <iomanip> #include <fstream>  using namespace std;  class Student { protected:  string name;  string rollNumber;  string fatherName;  int age;  int subjects;  string email; //declaring  string gender;  long int phoneNumber;  int semester;  float cgpa;  float creditHoursOfCurrentSemester;  string address;  int quizzesAmount;  int assignmentAmount;  float MarksInAssignment;  float MarksInQuizzes;  float MarksInMid;  float MarksInFinal;  float retakeMidMarks;  int totalMarksOfQuiz;  int totalMarksOfAssignment;  int totalMarksOfMid;  int totalMarksOfFinal;  int totalMarksOfRetake; public:  Student() {  name = " ";  age = 0;  rollNumber = " ";  fatherName = " ";  subjects = 0; //setting them to zero in default parameter  email = " ";  gender = " ";  phoneNumber = 00;  semester = 0;  cgpa = 0.00;  creditHoursOfCurrentSemester = 0.00;  address = " ";  quizzesAmount=0;  assignmentAmount=0;  MarksInAssignment=0.00;  MarksInFinal=0.00;  MarksInMid=0.00;  MarksInQuizzes=0.00;  retakeMidMarks=0.00;  totalMarksOfAssignment=0;  totalMarksOfFinal=0;  totalMarksOfMid=0;  totalMarksOfQuiz=0;  totalMarksOfRetake=0;  }   Student(string N, string Address, int Age, string RN, string fN, int subj, string Email, string Gender,  long int Pn, int Semester, float Cgpa, float cHOfCSemester,int qA,int aA,float MInAssig,float MInQuiz,float MInFinal,float MInMid,float rTMidMarks,int tMOfAssig,int tMOfQuiz,int tMOfMid,int tMOfFinal,int tMOfRetake) {  name = N;  age = Age;  rollNumber = RN;  fatherName = fN;  subjects = subj;  email = Email;  gender = Gender;  phoneNumber = Pn;  semester = Semester; //setting the values in parametrized constructor  cgpa = Cgpa;  creditHoursOfCurrentSemester = cHOfCSemester;  address = Address;  quizzesAmount=qA;  assignmentAmount=aA;  MarksInAssignment=MInAssig;  MarksInFinal=MInFinal;  MarksInMid=MInMid;  MarksInQuizzes=MInQuiz;  retakeMidMarks=rTMidMarks;  totalMarksOfAssignment=tMOfAssig;  totalMarksOfQuiz=tMOfQuiz;  totalMarksOfFinal=tMOfFinal;  totalMarksOfMid=tMOfMid;  totalMarksOfRetake=tMOfRetake;  }   void getStudentInfo() {  cout << "Enter Name : ";  cin.ignore();  getline(cin, name); //taking name  cout << "Enter Age = ";  cin >> age; //taking age  cout << "Enter gender : ";  cin.ignore(); //taking personal info from the user  getline(cin, gender);  cout << "Enter Father name : ";  getline(cin, fatherName);  cout << "Enter Phone Number = ";  cin >> phoneNumber;  cout << "Enter email : ";  cin.ignore();  getline(cin, email);  cout << "Enter address : ";  getline(cin, address);  }   void getAcademicInfo() {  cout << "Enter roll number : ";  cin.ignore();  getline(cin, rollNumber); //taking roll number  cout << "How many subjects are you studying in this semester = ";  cin >> subjects;  cout << "Enter current semester = "; //taking academic info from the user  cin >> semester;  cout << "Enter cgpa = ";  cin >> cgpa;  cout << "Enter number of credit hours = ";  cin >> creditHoursOfCurrentSemester; //taking credit hours  }  void getMarksInfo(){  int opt;  cout<<"Enter total number Of Quizzes = ";  cin>>quizzesAmount; //taking marks info from the user  cout<<"Enter number Of Assignment = ";  cin>>assignmentAmount;  cout<<"ENTER TOTAL MARKS OF QUIZ = ";  cin>>totalMarksOfQuiz;  cout<<"ENTER TOTAL MARKS OF ASSIGNMENT = ";  cin>>totalMarksOfAssignment;  cout<<"ENTER TOTAL MARKS OF MID = ";  cin>>totalMarksOfMid;  cout<<"ENTER TOTAL MARKS OF RETAKE MID = ";  cin>>totalMarksOfRetake;  cout<<"ENTER TOTAL MARKS OF FINAL EXAM = ";  cin>>totalMarksOfFinal;  cout<<"Enter quiz marks = ";  for(int i=0;i<quizzesAmount;i++){  cin>>MarksInQuizzes; //taking marks of the quiz in loop  if(MarksInQuizzes>totalMarksOfQuiz){  cout<<"you have entered invalid number.."<<endl;  cout<<"OBTAIN MARKS CAN NOT BE GRATER THAN TOTAL MARKS.."<<endl; //after checking that the obtain marks is not greater than total marks exiting  exit(2);  }  }  cout<<"Enter assignment marks = ";  for(int i=0;i<assignmentAmount;i++){  cin>>MarksInAssignment; //taking assignment marks  if(MarksInAssignment>totalMarksOfAssignment){  cout<<"you have entered invalid number.."<<endl;  cout<<"OBTAIN MARKS CAN NOT BE GRATER THAN TOTAL MARKS.."<<endl;  exit(2);  }  }  cout<<"Enter Mid Exam marks = ";  cin>>MarksInMid; //taking mid marks  if(MarksInMid>totalMarksOfMid){  cout<<"you have entered invalid number.."<<endl;  cout<<"OBTAIN MARKS CAN NOT BE GRATER THAN TOTAL MARKS.."<<endl;  exit(2);  }  cout<<"Enter Final Exam marks = ";  cin>>MarksInFinal; //taking final marks  if(MarksInFinal>totalMarksOfFinal){  cout<<"you have entered invalid number.."<<endl;  cout<<"OBTAIN MARKS CAN NOT BE GRATER THAN TOTAL MARKS.."<<endl;  exit(2);  }  cout<<"WAS THERE ANY RETAKE MID?"<<endl;  cout<<"1.yes."<<endl;  cout<<"2.no."<<endl; //checking that if the retake is taken or not  cin>>opt;  if(opt==1){  cout<<"Enter Retake Mid Exam marks = ";  cin>>retakeMidMarks;  if(retakeMidMarks>totalMarksOfRetake){  cout<<"you have entered invalid number.."<<endl;  cout<<"OBTAIN MARKS CAN NOT BE GRATER THAN TOTAL MARKS.."<<endl;  exit(2);  }  }  if(opt==2){  cout<<"THERE WASN'T ANY RETAKE MID EXAM.."<<endl;  }  }  void displayMarksInfo(){  fstream File;  File.open("studentMarksInfo.txt", ios::app);  if (File.is\_open()) {  for(int i=0;i<quizzesAmount;i++){  File<<"quiz marks "<<MarksInQuizzes<<setw(20)<<"out of"<<setw(10)<<totalMarksOfQuiz<<setw(20);  } //storing in file  cout<<endl;  for(int i=0;i<quizzesAmount;i++){  File<<"quiz marks "<<MarksInQuizzes<<setw(20)<<"out of "<<setw(10)<<totalMarksOfAssignment<<setw(20);  }  cout<<endl;  File<<"MID EXAM MARKS = "<<MarksInMid<<setw(20)<<"out of"<<setw(10)<<totalMarksOfMid<<endl;  File<<"RETAKE MID EXAM = "<<retakeMidMarks<<setw(20)<<"out of"<<setw(10)<<totalMarksOfRetake<<endl;  if(retakeMidMarks>MarksInMid){  File<<"FINAL MID TERM EXAM MARKS = "<<retakeMidMarks<<setw(20)<<"out of"<<setw(10)<<totalMarksOfRetake<<endl;  }  else{  File<<"MID EXAM MARKS = "<<MarksInMid<<setw(20)<<"out of"<<setw(10)<<totalMarksOfMid<<endl;  }  File<<"FINAL EXAM MARKS = "<<MarksInFinal<<setw(20)<<"out of"<<setw(10)<<totalMarksOfFinal<<endl;  }  else{  cout<<"FILE IS NOT OPEN.. "<<endl;  }  File.close();  }    void displayStudentPersonalInfo() {  fstream file;  file.open("studentPersonalInfo.txt", ios::app);  if (file.is\_open()) {  file << "Name : " << name << setw(10) << "Age = " << age << setw(10) << endl;  file << "Father name : " << fatherName << setw(10) << "Email : " << email << setw(10) << endl;  file << "Gender : " << gender << setw(10) << " Phone Number = " << phoneNumber << setw(10) << endl;  file << "Address = " << address << setw(10) << endl; //storing in file  file << "------------------------------------------------------------------------------------" << endl;  } else {  cout << "FILE IS NOT OPEN.." << endl;  }  file.close();   }   void displayAcademicInfo() {  fstream file1;  file1.open("studentAcademicInfo.txt", ios::app);  if (file1.is\_open()) {  file1 << "Roll number : " << rollNumber << setw(10) << "Number Of Subjects In Current Semester= " << subjects  << setw(10) << endl; //storing in file  file1 << "Current semester = " << semester << setw(10) << "cgpa = " << cgpa << setw(10) << endl;  file1 << "credit hours of current semester = " << creditHoursOfCurrentSemester << setw(10) << endl;  file1 << "------------------------------------------------------------------------------------" << endl;  } else {  cout << "FILE IS NOT OPEN.." << endl;  }  file1.close();  } };  class universityManagementSystem { private:  int numberOfStudentsInClass; //declaring  public:  Student S; //making object   universityManagementSystem() {  numberOfStudentsInClass = 0; //setting the value to zero  }   universityManagementSystem(int numberOfStudent) {  numberOfStudentsInClass = numberOfStudent; //equaling  }   void displayPersonalInfoOfStudent() {  cout<<"ENTER NUMBER OF STUDENTS = ";  cin >> numberOfStudentsInClass;  for (int i = 0; i < numberOfStudentsInClass; i++) { //loop at number of students  S.getStudentInfo(); //calling functions  S.displayStudentPersonalInfo();  }  }   void displayAcademicInfoOfStudent() {  cout<<"ENTER NUMBER OF STUDENTS = ";  cin >> numberOfStudentsInClass;  for (int i = 0; i < numberOfStudentsInClass; i++) {  S.getAcademicInfo(); //calling functions  S.displayAcademicInfo();  }  }  void displayMarksInfo(){  cout<<"ENTER NUMBER OF STUDENTS = ";  cin>>numberOfStudentsInClass;  for (int i = 0; i < numberOfStudentsInClass; i++) {  S.getMarksInfo(); //calling functions  S.displayMarksInfo();  }  }  };  **Main.cpp:**  #include <iostream> #include "Functions.h" using namespace std;  int main() {  int opt;  universityManagementSystem U;  universityManagementSystem U1(1);  do{  cout<<"CHOOSE ANY OPTION."<<endl;  cout<<"1.DISPLAY PERSONAL INFORMATION."<<endl;  cout<<"2.DISPLAY ACADEMIC INFORMATION."<<endl; //taking option  cout<<"3.DISPLAY MARKS INFO."<<endl;  cout<<"4.exit.."<<endl;  cin>>opt;  if(opt==1){  U1.displayPersonalInfoOfStudent(); //calling  }  else if(opt==2){  U1.displayAcademicInfoOfStudent(); //calling  }  else if(opt==3){  U1.displayMarksInfo(); //calling  }  else if(opt==4){  cout<<"YOU CHOOSE TO EXIT.."<<endl; //exit statement  exit(3);  }  else{  cout<<"YOU HAVE ENTERED AN INVALID ARGUMENT.."<<endl;  exit(3); //exit statement  }  }while(opt!=4);   return 0; }  **Output:**   * It is stored in file… |

#### **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** | **Task** | **CLO** | **Description** | **Max marks** | **Exceeds expectation** | **Meets expectation** | **Does not meet expectation** | **Obtained marks** |
| 1. Realization of experiment (a) | 1 | 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 | 3 | 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 | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 1 | 1 | 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 | 3 | Code commenting | 5 | Comments are added and does help the reader to understand the code (4-5) | Comments are added and does not help the reader to understand the code (2-3) | Comments are not added (0-1) |  |
| 5. Data collection (c) | 1 | 3 | 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 | 4 | 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 | 2 | Documentation & GitHub Submissions | 5 | Timely (4-5) | Late (2-3) | Not done (0-1) |  |
|  | Max Marks (total): | | | 100 | Obtained Marks (total): | | |  |

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