Title: Examination System for Course Evaluations

Introduction:

The Examination System for Course Evaluations project aims to design and implement a comprehensive system that facilitates instructors in setting, conducting, marking, and preparing results for course evaluations. The system caters to two user roles: students and teachers. A user-friendly menu interface is provided, offering general information accessible to all users. Upon logging in, students gain access to additional functionalities such as viewing enrolled courses, changing passwords, accessing pending quizzes, and checking their marks. Conversely, teachers can view course-specific information, including course names, codes, and the list of enrolled students. They possess additional abilities such as setting quizzes and utilizing quiz analytics to assess question effectiveness. Attendance sheets are also generated to track students' quiz attempts. Additionally, a question bank with multiple-choice, true/false, and descriptive questions is created, aligned with course topics, and stored in a CSV file.

Methodology:

The Examination System for Course Evaluations project was developed using the principles of Object-Oriented Programming (OOP) and implemented in the C++ programming language.

One of the initial challenges encountered during the implementation was reading and writing data from CSV files using C++. Extensive research and reference to online resources were undertaken to understand the necessary syntax and libraries required to handle CSV file operations effectively.

Another significant challenge was designing the overall structure of the project. To address this, a comprehensive analysis of the project requirements and available resources was conducted. Online resources and relevant documentation were consulted to gather insights into best practices and design patterns for building modular and scalable systems. The acquired knowledge was then applied to create an organized and cohesive design that separates concerns, maximizes code reuse, and promotes maintainability.

System Design and Architecture:

The Examination System for Course Evaluations follows a well-defined system design and architecture that promotes modularity, reusability, and maintainability. The system architecture can be visualized using the UML class diagram.

At the core of the architecture is the "Display" class, which serves as a central component responsible for displaying information and coordinating interactions between different classes.

This design choice ensures a cohesive user experience and facilitates the retrieval of relevant data from the appropriate classes.

The system employs an object-oriented approach, with the "User" class acting as a parent class and two child classes, "Student" and "Teacher," inheriting from it. This inheritance hierarchy allows for the specialization of functionalities and attributes specific to each user role, providing a clear separation of concerns.

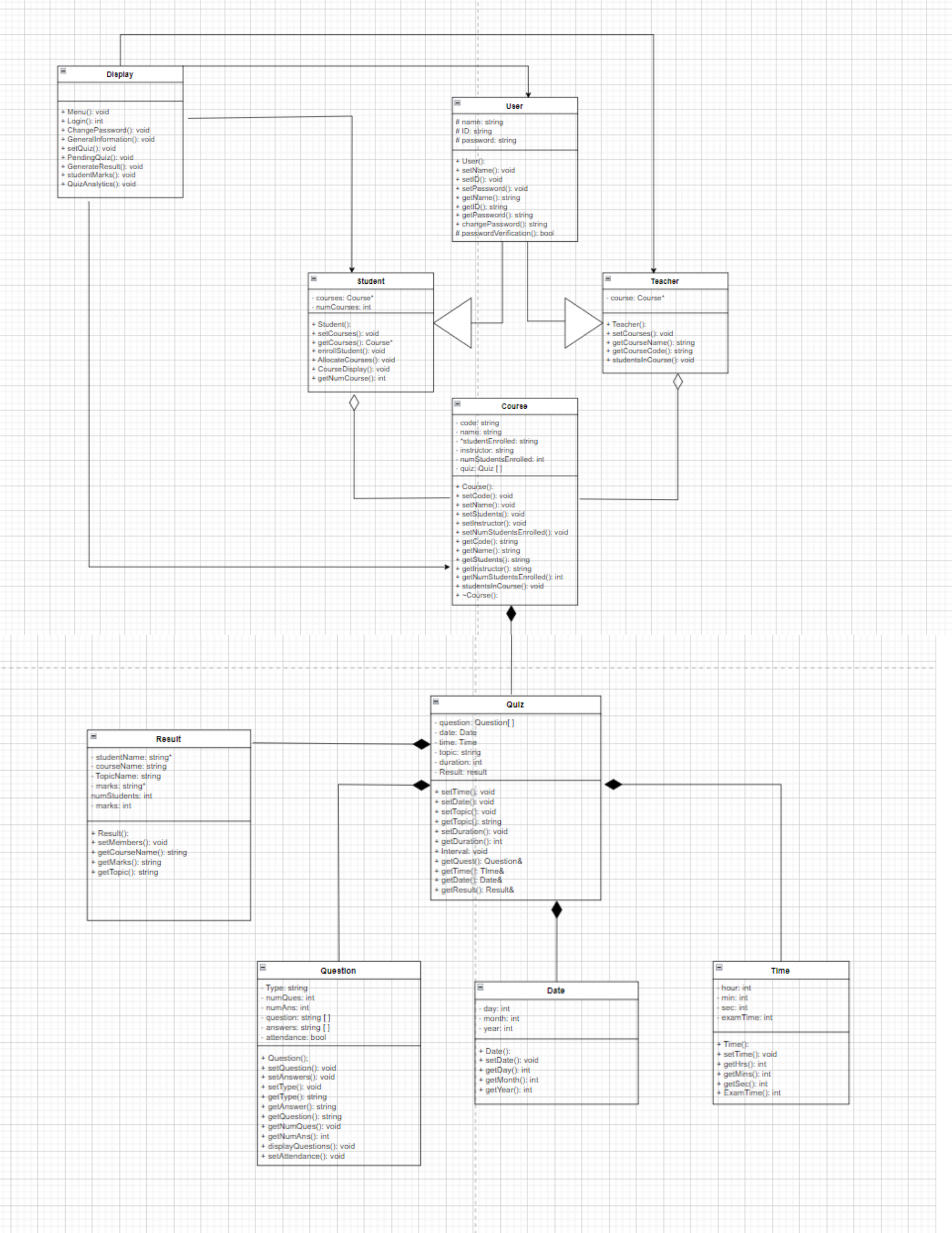
The "Course" class, independent of the user classes, represents a course entity. It is associated with both "Teacher" and "Student" classes through aggregation. This association enables the linkage of students and teachers to their respective courses, allowing for efficient management and retrieval of course-related information.

Within the "Course" class, there is a composition relationship with the "Quiz" class. This composition signifies that a course can have multiple quizzes, each represented by an instance of the "Quiz" class. By composing the "Quiz" class with the "Course" class, the system ensures the encapsulation of quiz-specific functionalities and facilitates easy navigation between courses and quizzes.

The "Quiz" class itself composes four additional classes: "Question," "Time," "Date," and "Result." This composition relationship allows for the organization and management of quiz-related data. Each question within a quiz is represented by an instance of the "Question" class, with additional classes such as "Time" and "Date" capturing temporal information, and the "Result" class storing the quiz outcome.

By employing this system design and architecture, the Examination System for Course Evaluations achieves a modular and scalable structure. The separation of concerns, clear class relationships, and encapsulation of functionalities contribute to a well-organized and efficient system that effectively addresses the requirements of course evaluations.

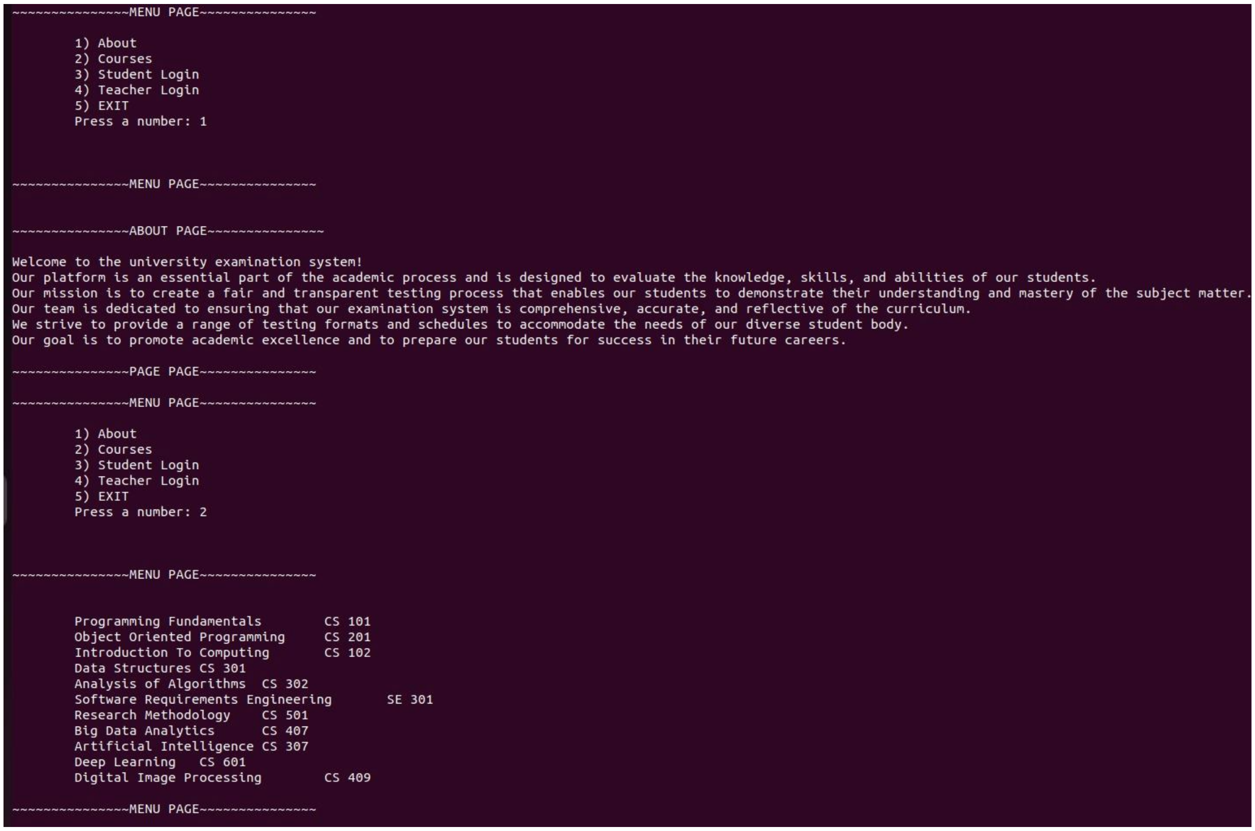
The Class Diagram is shown below:


Display
+ Menu(): void
+ Login(): int
+ ChangePassword(): void
+ GeneralInformation(): void
+ setQuiz(): void
+ PendingQuiz(): void
+ GenerateResult(): void
+ studentMarks(): void
+ QuizAnalytics(): void
User
# name: string
# ID: string
# password: string
+ User():
+ setName(): void
+ setID(): void
+ setPassword(): void
+ getName(): string
+ getID(): string
+ getPassword(): string
+ changePassword(): string
# passwordVerification(): bool
Student
- courses: Course*
- numCourses: int
+ Student():
+ setCourses(): void
+ getCourses(): Course*
+ enrollStudent(): void
+ AllocateCourses(): void
+ CourseDisplay(): void
+ getNumCourse(): int
Teacher
- course: Course*
+ Teacher():
+ setCourses(): void
+ getCourseName(): string
+ getCourseCode(): string
+ studentsInCourse(): void
Course
- code: string
- name: string
- *studentEnrolled: string
- instructor: string
- numStudentsEnrolled: int
- quiz: Quiz [ ]
+ Course():
+ setCode(): void
+ setName(): void
+ setStudents(): void
+ setInstructor(): void
+ setNumStudentsEnrolled(): void
+ getCode(): string
+ getName(): string
+ getStudents(): string
+ getInstructor(): string
+ getNumStudentsEnrolled(): int
+ studentsInCourse(): void
+ ~Course():
Quiz
- question: Question[ ]
- date: Date
- time: Time
- topic: string
- duration: int
- Result: result
+ setTime(): void
+ setDate(): void
+ setTopic(): void
+ getTopic(): string
+ setDuration(): void
+ getDuration(): int
+ Interval: void
+ getQuest(): Question&
+ getTime(): TIme&
+ getDate(): Date&
+ getResult(): Result&
Result
- studentName: string*
- courseName: string
- TopicName: string
- marks: string*
numStudents: int
- marks: int
+ Result():
+ setMembers(): void
+ getCourseName(): string
+ getMarks(): string
+ getTopic(): string
Question
- Type: string
- numQues: int
- numAns: int
- question: string [ ]
- answers: string [ ]
- attendance: bool
+ Question():
+ setQuestion(): void
+ setAnswers(): void
+ setType(): void
+ getType(): string
+ getAnswer(): string
+ getQuestion(): string
+ getNumQues(): void
+ getNumAns(): int
+ displayQuestions(): void
+ setAttendance(): void
Date
- day: int
- month: int
- year: int
+ Date():
+ setDate(): void
+ getDay(): int
+ getMonth(): int
+ getYear(): int
Time
- hour: int
- min: int
- sec: int
- examTime: int
+ Time():
+ setTime(): void
+ getHrs(): int
+ getMins(): int
+ getSec(): int
+ ExamTime(): int

Functionality:

First of all, I created 216 objects of Students, 11 objects of Teachers, and 11 Objects of Courses. Then, I read the three CSV files, which include Students Information, Teacher information, and Course information, and stored the data in their respective Objects. Then I read a Quiz CSV file that has questions and their answers stored according to the topics and stored them in the Quiz object that was composed in the respective Courses. All of this was done in the main function.

Then, I created an object for the Display class and called the member function named menu.of this class. I then displayed a menu for a General User, Student, and Teacher. This can be seen as shown below:

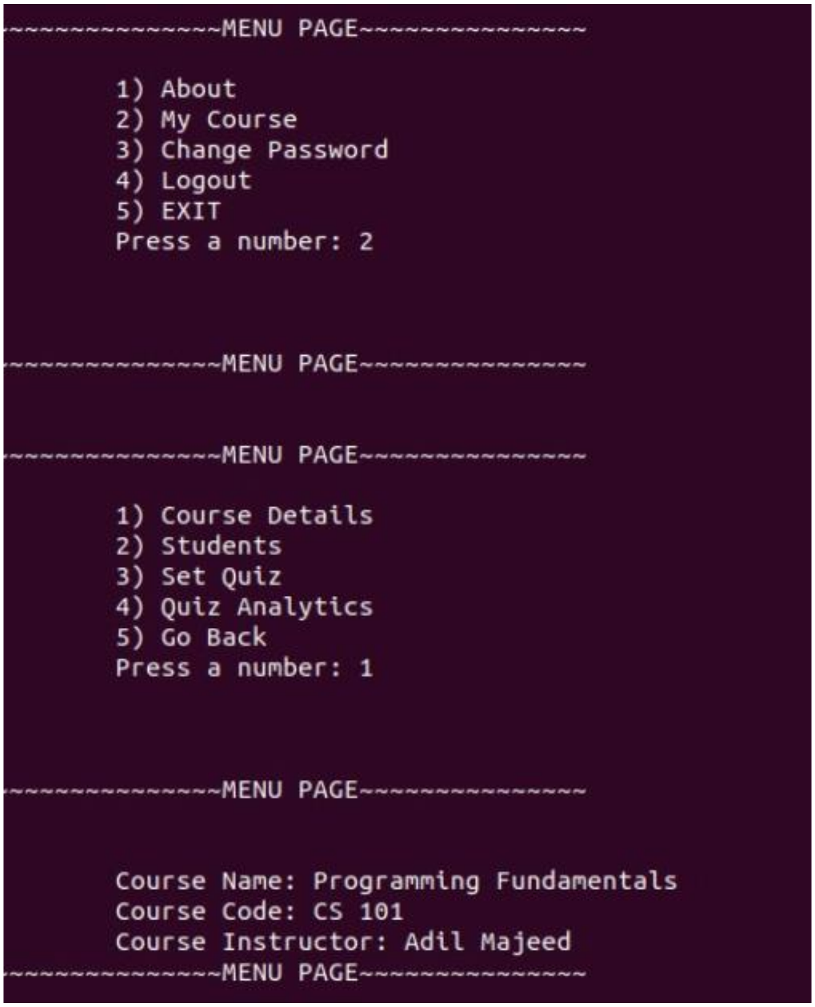


```
~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) Courses
        3) Student Login
        4) Teacher Login
        5) EXIT
        Press a number: 1



~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


~~~~~~~~~~~~~~~ABOUT PAGE~~~~~~~~~~~~~~~

Welcome to the university examination system!
Our platform is an essential part of the academic process and is designed to evaluate the knowledge, skills, and abilities of our students.
Our mission is to create a fair and transparent testing process that enables our students to demonstrate their understanding and mastery of the subject matter.
Our team is dedicated to ensuring that our examination system is comprehensive, accurate, and reflective of the curriculum.
We strive to provide a range of testing formats and schedules to accommodate the needs of our diverse student body.
Our goal is to promote academic excellence and to prepare our students for success in their future careers.

~~~~~~~~~~~~~~~PAGE PAGE~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) Courses
        3) Student Login
        4) Teacher Login
        5) EXIT
        Press a number: 2



~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


        Programming Fundamentals        CS 101
        Object Oriented Programming     CS 201
        Introduction To Computing       CS 102
        Data Structures CS 301
        Analysis of Algorithms  CS 302
        Software Requirements Engineering       SE 301
        Research Methodology    CS 501
        Big Data Analytics      CS 407
        Artificial Intelligence CS 307
        Deep Learning   CS 601
        Digital Image Processing        CS 409

~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~
```

I also gave an option of About, which displays General information about the corresponding Examination system. For this, I made a member function of Display Class named General Information. I also displayed an option of the courses available and displayed the available courses by accessing the teachers' 11 objects and through it accessing its member functions of getCourseName and getCourseCode.

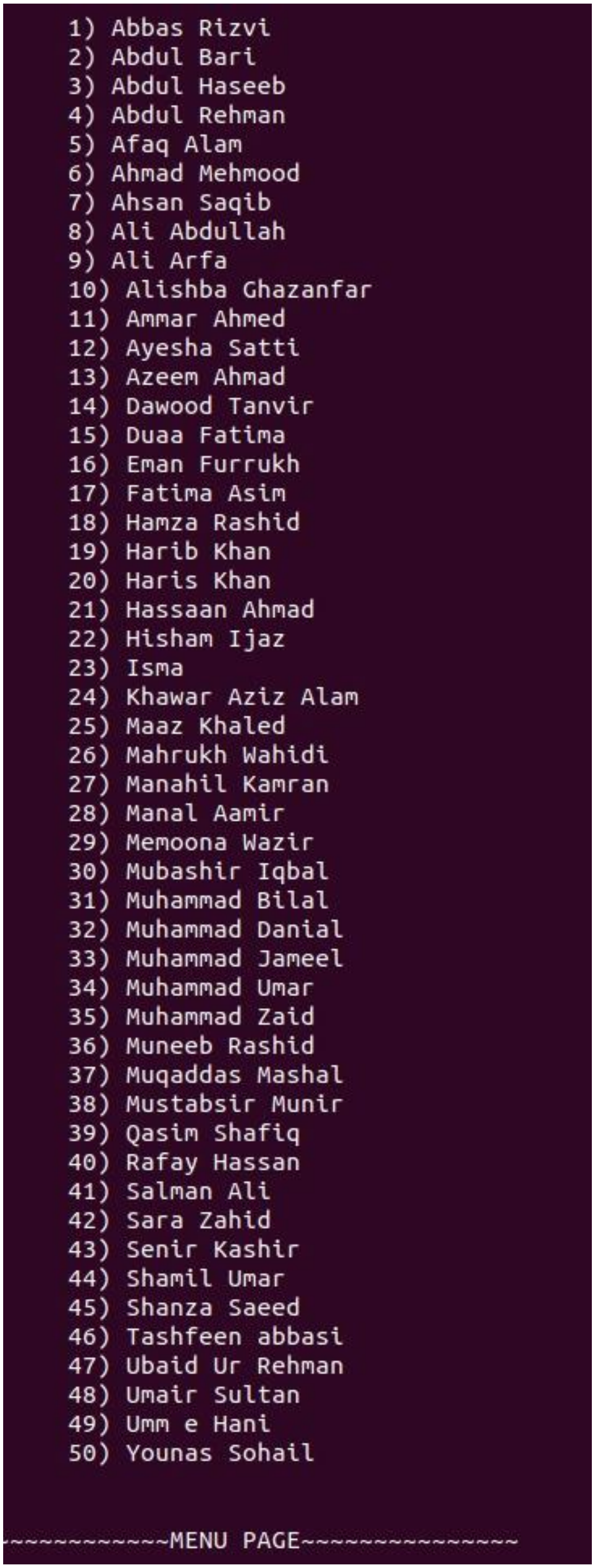
After that, I made a member function of the Display class named Login Function, which is able to Login both students and teachers when pressed 3 or 4.

When logged in, for instance, a teacher logs in, then they have the following options:



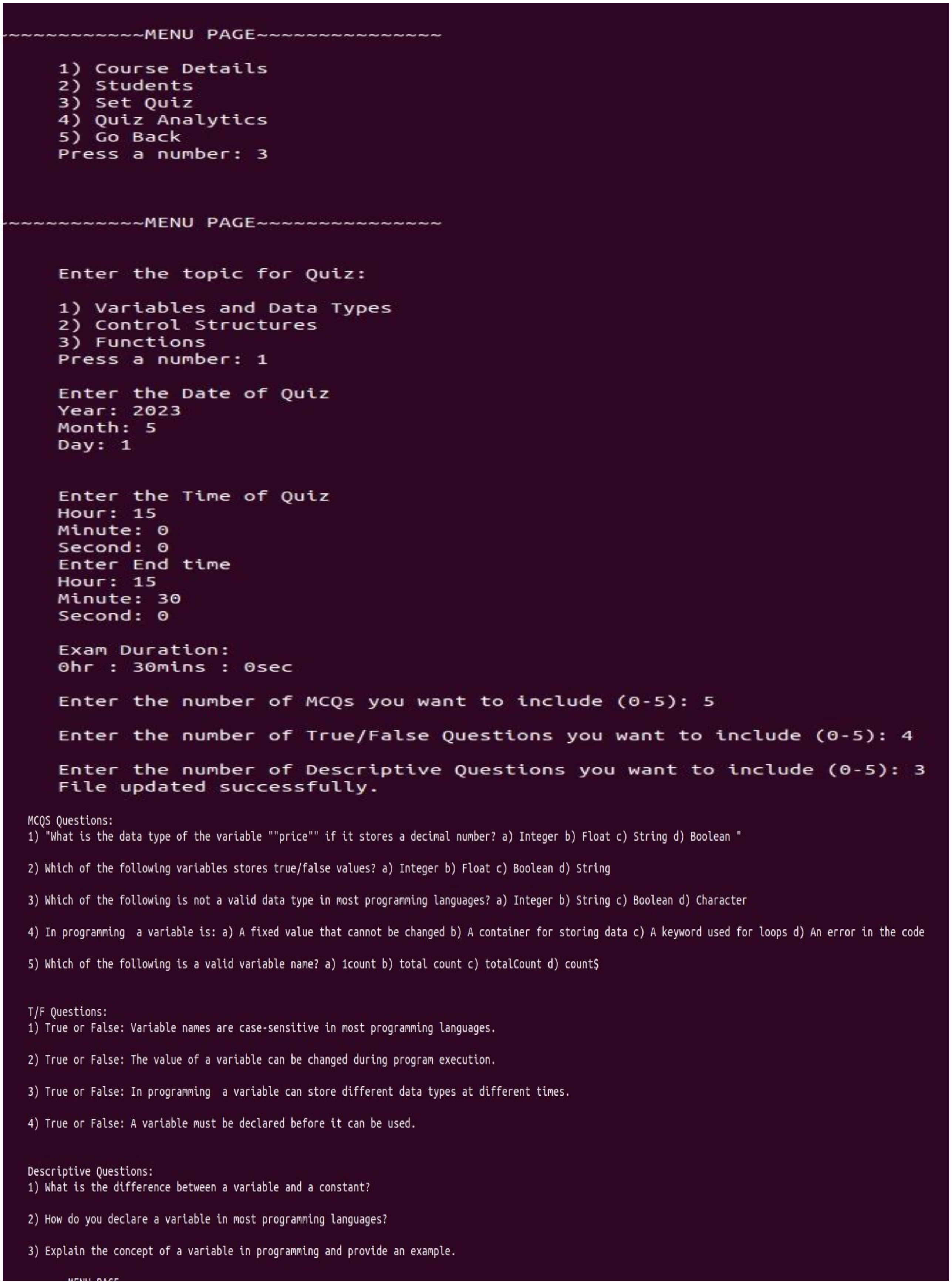
```
~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

       1) About
       2) My Course
       3) Change Password
       4) Logout
       5) EXIT
       Press a number: 2



~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

       1) Course Details
       2) Students
       3) Set Quiz
       4) Quiz Analytics
       5) Go Back
       Press a number: 1



~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


       Course Name: Programming Fundamentals
       Course Code: CS 101
       Course Instructor: Adil Majeed
~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~
```

When pressed 2, the window redirects to another menu that displays specific information of the Course of the teacher which can be shown above. After that, when the teacher presses 1), the program displays the course details of that teacher's course, that are, course name, course code and course instructor. This is done by accessing the member function of the Teacher's class, namely, getCourseName, getCourseCode, and getName.



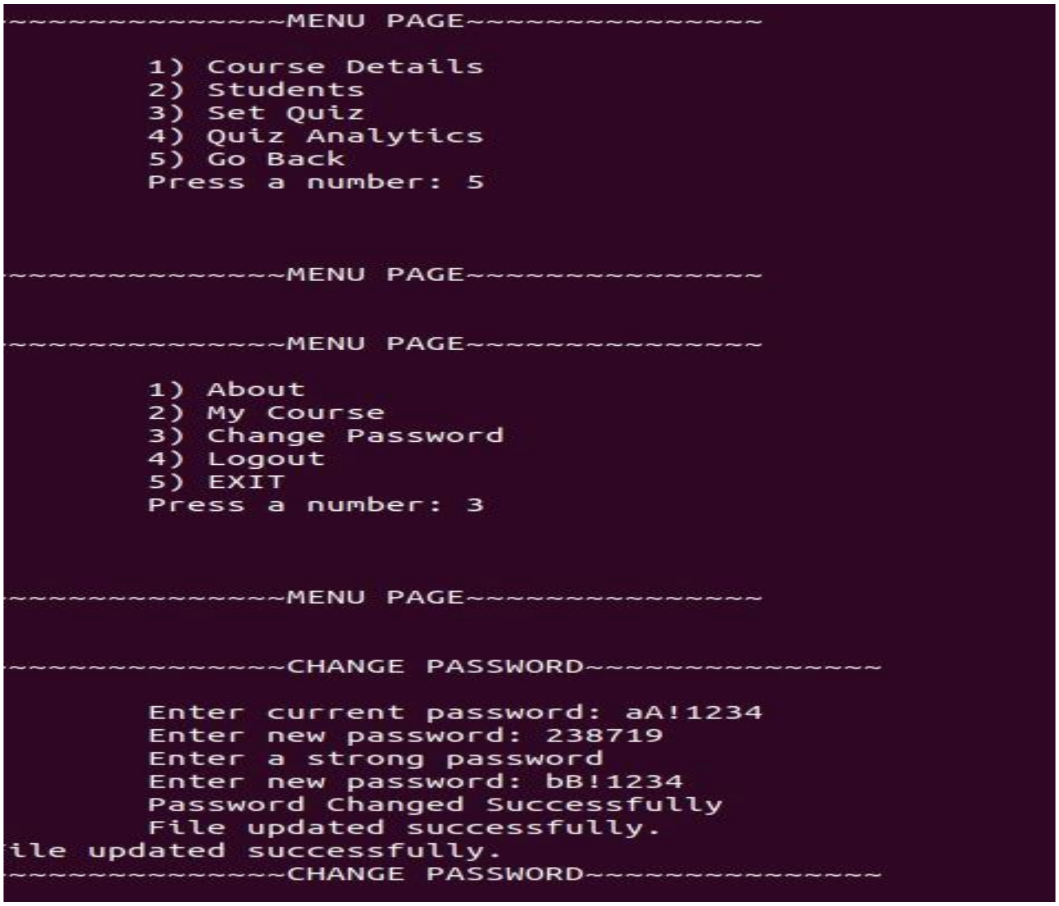
```
     1) Abbas Rizvi
     2) Abdul Bari
     3) Abdul Haseeb
     4) Abdul Rehman
     5) Afaq Alam
     6) Ahmad Mehmood
     7) Ahsan Saqib
     8) Ali Abdullah
     9) Ali Arfa
     10) Alishba Ghazanfar
     11) Ammar Ahmed
     12) Ayesha Satti
     13) Azeem Ahmad
     14) Dawood Tanvir
     15) Duaa Fatima
     16) Eman Furrukh
     17) Fatima Asim
     18) Hamza Rashid
     19) Harib Khan
     20) Haris Khan
     21) Hassaan Ahmad
     22) Hisham Ijaz
     23) Isma
     24) Khawar Aziz Alam
     25) Maaz Khaled
     26) Mahrukh Wahidi
     27) Manahil Kamran
     28) Manal Aamir
     29) Memoona Wazir
     30) Mubashir Iqbal
     31) Muhammad Bilal
     32) Muhammad Danial
     33) Muhammad Jameel
     34) Muhammad Umar
     35) Muhammad Zaid
     36) Muneeb Rashid
     37) Muqaddas Mashal
     38) Mustabsir Munir
     39) Qasim Shafiq
     40) Rafay Hassan
     41) Salman Ali
     42) Sara Zahid
     43) Senir Kashir
     44) Shamil Umar
     45) Shanza Saeed
     46) Tashfeen abbasi
     47) Ubaid Ur Rehman
     48) Umair Sultan
     49) Umm e Hani
     50) Younas Sohail


~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~
```

After that, when pressed 2, the program displays the names of the students enrolled in the corresponding course which can also be seen. This is displayed through a member function of the teacher class named StudentsInCourse, which displays the names of the students available in the course.



```
~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

    1) Course Details
    2) Students
    3) Set Quiz
    4) Quiz Analytics
    5) Go Back
    Press a number: 3



~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


    Enter the topic for Quiz:

    1) Variables and Data Types
    2) Control Structures
    3) Functions
    Press a number: 1

    Enter the Date of Quiz
    Year: 2023
    Month: 5
    Day: 1


    Enter the Time of Quiz
    Hour: 15
    Minute: 0
    Second: 0
    Enter End time
    Hour: 15
    Minute: 30
    Second: 0

    Exam Duration:
    0hr : 30mins : 0sec

    Enter the number of MCQs you want to include (0-5): 5

    Enter the number of True/False Questions you want to include (0-5): 4

    Enter the number of Descriptive Questions you want to include (0-5): 3
    File updated successfully.

MCQS Questions:
1) "What is the data type of the variable ""price"" if it stores a decimal number? a) Integer b) Float c) String d) Boolean "

2) Which of the following variables stores true/false values? a) Integer b) Float c) Boolean d) String

3) Which of the following is not a valid data type in most programming languages? a) Integer b) String c) Boolean d) Character

4) In programming  a variable is: a) A fixed value that cannot be changed b) A container for storing data c) A keyword used for loops d) An error in the code

5) Which of the following is a valid variable name? a) 1count b) total count c) totalCount d) count$


T/F Questions:
1) True or False: Variable names are case-sensitive in most programming languages.

2) True or False: The value of a variable can be changed during program execution.

3) True or False: In programming  a variable can store different data types at different times.

4) True or False: A variable must be declared before it can be used.


Descriptive Questions:
1) What is the difference between a variable and a constant?

2) How do you declare a variable in most programming languages?

3) Explain the concept of a variable in programming and provide an example.

        MENU PAGE
```

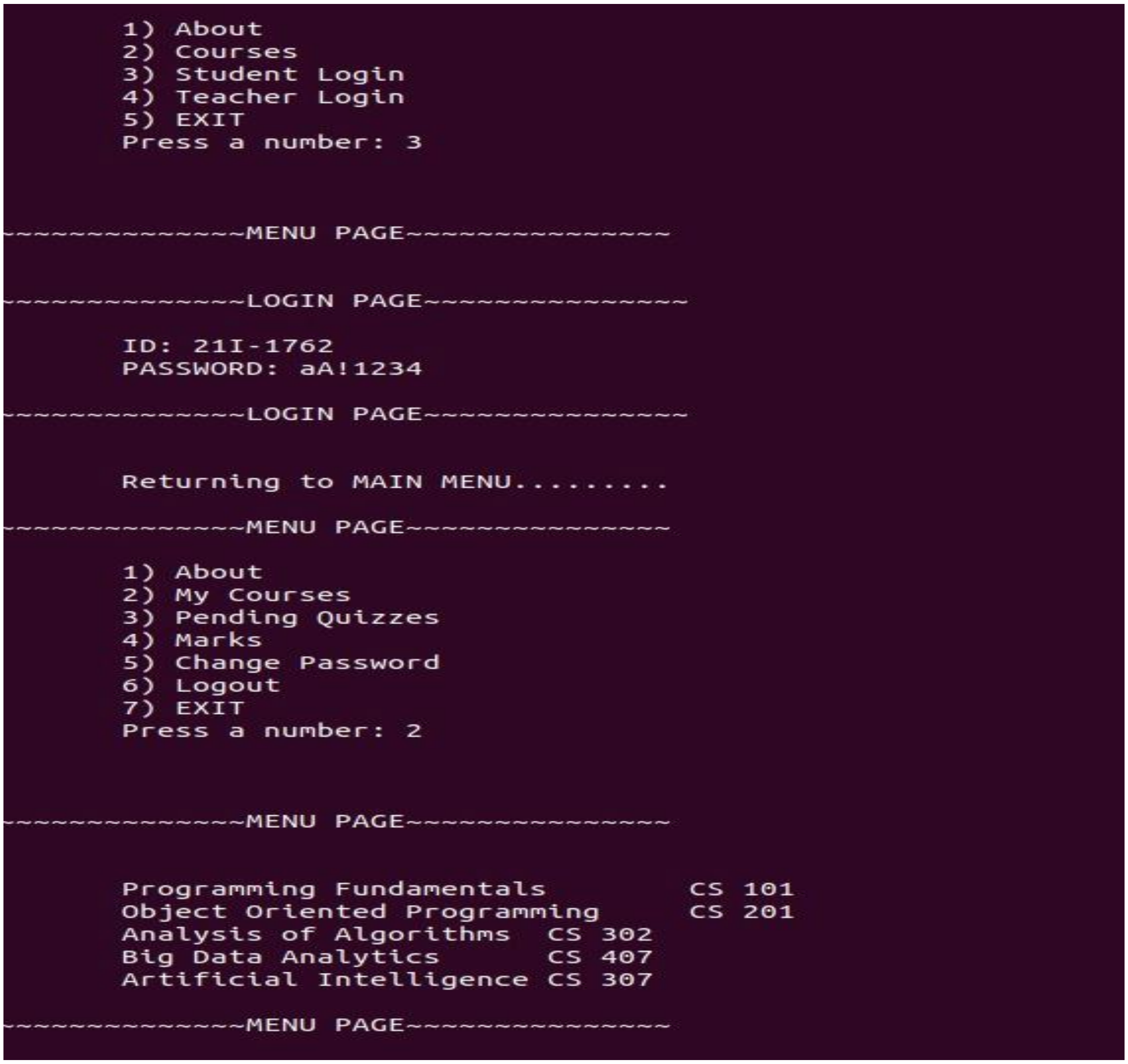
By pressing 3, the teacher can set a Quiz of their own course. Each course has 3 topics and these topics have further 5 MCQs, True/False Questions, and Descriptive Questions which I had already stored at the start of the program in the Quiz class. By pressing 3, the program them calls a setQuiz member function of the display class, which asks for the topic name, start time, end time, number of MCQs, True/False and Descriptive Questions and then stores the data in a new file named PendingQuiz.csv. After setting the quiz, the quiz is displayed to the teacher by shuffling the position of the questions.



```
~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

       1) Course Details
       2) Students
       3) Set Quiz
       4) Quiz Analytics
       5) Go Back
       Press a number: 5



~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

       1) About
       2) My Course
       3) Change Password
       4) Logout
       5) EXIT
       Press a number: 3



~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


~~~~~~~~~~~~~~CHANGE PASSWORD~~~~~~~~~~~~~~~

       Enter current password: aA!1234
       Enter new password: 238719
       Enter a strong password
       Enter new password: bB!1234
       Password Changed Successfully
       File updated successfully.
ile updated successfully.
~~~~~~~~~~~~~~CHANGE PASSWORD~~~~~~~~~~~~~~~
```

The teacher and student both can change their passwords, when choosing this option, the program calls a member function of the Display class named ChangePassword, and uses it to change the password by validating it through the member function of the User class named changePassword. The password should have atleast a digit, a special, and an upper case and lower case letters. All of this is done in the changePassword Function. This is then updated in the corresponding CSV file of teachers or students.

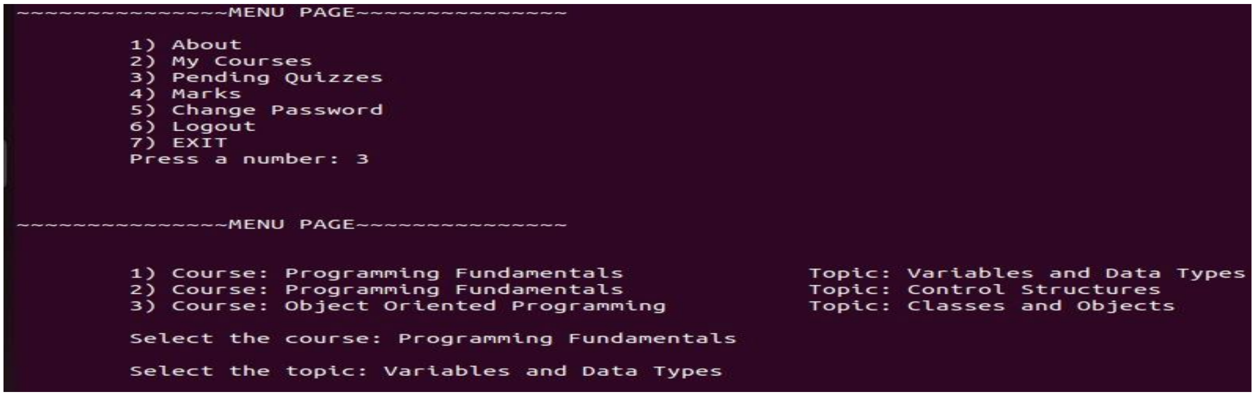
After that, logging in to student portal, the student have the following options:



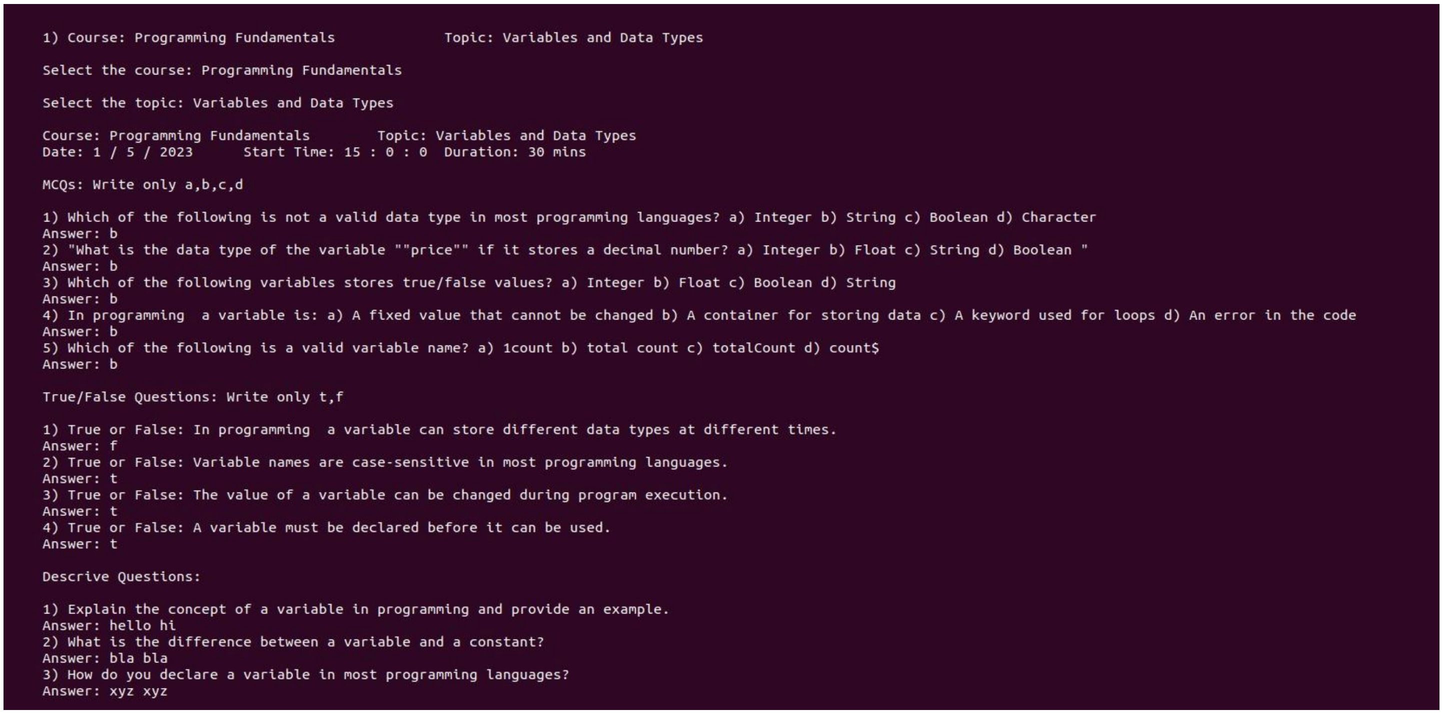
```
        1) About
        2) Courses
        3) Student Login
        4) Teacher Login
        5) EXIT
        Press a number: 3


~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


~~~~~~~~~~~~~~LOGIN PAGE~~~~~~~~~~~~~~~

        ID: 21I-1762
        PASSWORD: aA!1234

~~~~~~~~~~~~~~LOGIN PAGE~~~~~~~~~~~~~~~


        Returning to MAIN MENU.........

~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) My Courses
        3) Pending Quizzes
        4) Marks
        5) Change Password
        6) Logout
        7) EXIT
        Press a number: 2


~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


        Programming Fundamentals         CS 101
        Object Oriented Programming      CS 201
        Analysis of Algorithms  CS 302
        Big Data Analytics      CS 407
        Artificial Intelligence CS 307

~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~
```

The students can see the name of the courses they are enrolled in as well as the Course code. This is displayed through a member function of student class named CoursesDisplay.

After that, the student can choose option number 3) which redirects them to a menu of the available quizzes of the courses they are enrolled in as shown below

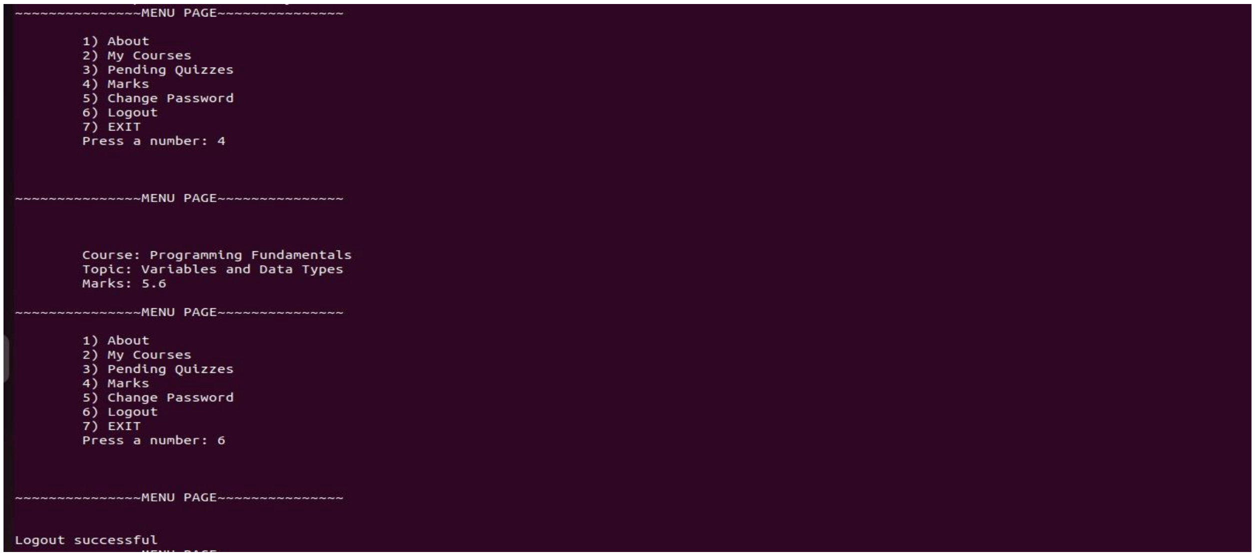


```
~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) My Courses
        3) Pending Quizzes
        4) Marks
        5) Change Password
        6) Logout
        7) EXIT
        Press a number: 3


~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


        1) Course: Programming Fundamentals            Topic: Variables and Data Types
        2) Course: Programming Fundamentals            Topic: Control Structures
        3) Course: Object Oriented Programming         Topic: Classes and Objects

        Select the course: Programming Fundamentals

        Select the topic: Variables and Data Types
```



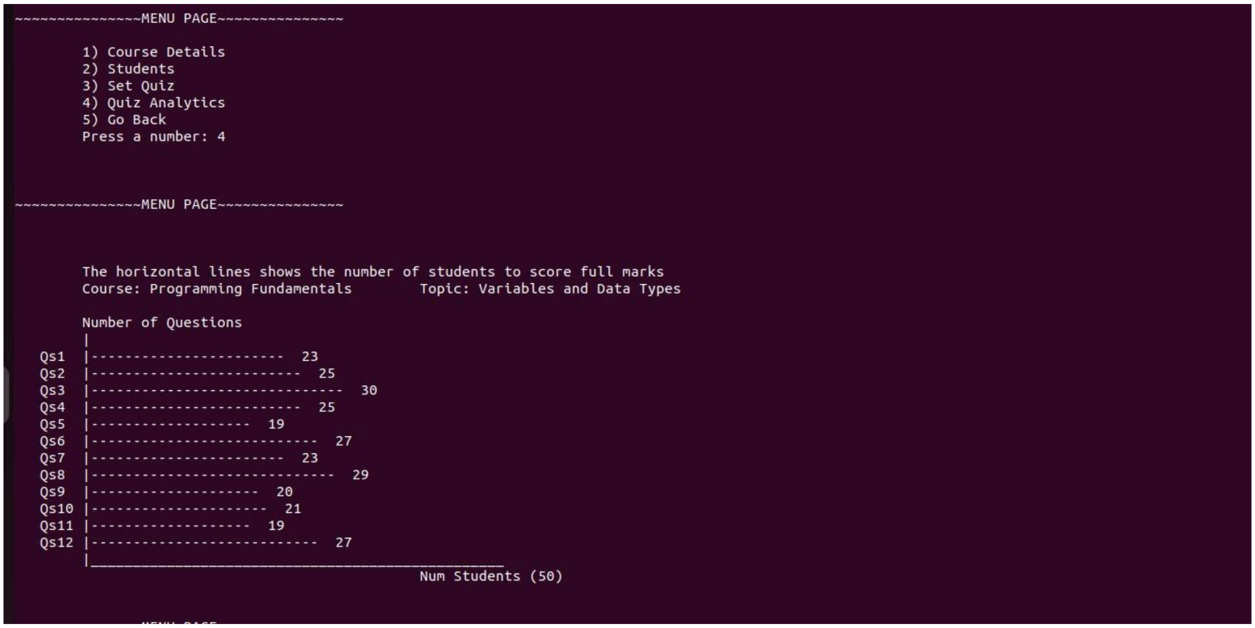
```
1) Course: Programming Fundamentals             Topic: Variables and Data Types

Select the course: Programming Fundamentals

Select the topic: Variables and Data Types

Course: Programming Fundamentals        Topic: Variables and Data Types
Date: 1 / 5 / 2023      Start Time: 15 : 0 : 0  Duration: 30 mins

MCQs: Write only a,b,c,d

1) Which of the following is not a valid data type in most programming languages? a) Integer b) String c) Boolean d) Character
Answer: b
2) "What is the data type of the variable ""price"" if it stores a decimal number? a) Integer b) Float c) String d) Boolean "
Answer: b
3) Which of the following variables stores true/false values? a) Integer b) Float c) Boolean d) String
Answer: b
4) In programming  a variable is: a) A fixed value that cannot be changed b) A container for storing data c) A keyword used for loops d) An error in the code
Answer: b
5) Which of the following is a valid variable name? a) 1count b) total count c) totalCount d) count$
Answer: b

True/False Questions: Write only t,f

1) True or False: In programming  a variable can store different data types at different times.
Answer: f
2) True or False: Variable names are case-sensitive in most programming languages.
Answer: t
3) True or False: The value of a variable can be changed during program execution.
Answer: t
4) True or False: A variable must be declared before it can be used.
Answer: t

Descrive Questions:

1) Explain the concept of a variable in programming and provide an example.
Answer: hello hi
2) What is the difference between a variable and a constant?
Answer: bla bla
3) How do you declare a variable in most programming languages?
Answer: xyz xyz
```

All of the functionality of attempting quiz and showing available quizzes is done inside the member function of the Display class named PendingQuiz, where data is read from the PendingQuiz.csv file generated earlier and stored inside different string arrays. The student then attempts the quiz and then another function is called named GenerateResult where the answers are compared from the answer key and the result is stored in MarkSheet.csv. Also, their attendance is marked too. After that, a member function of Result class is called which is named setMembers in the GenerateResult function. Here the program reads the Marksheet.csv, and stores the TopicName, courseName and Marks of the students which is derived from the own class of Student. Therefore, when a user presses the marks option in their main menu, the following output is displayed:



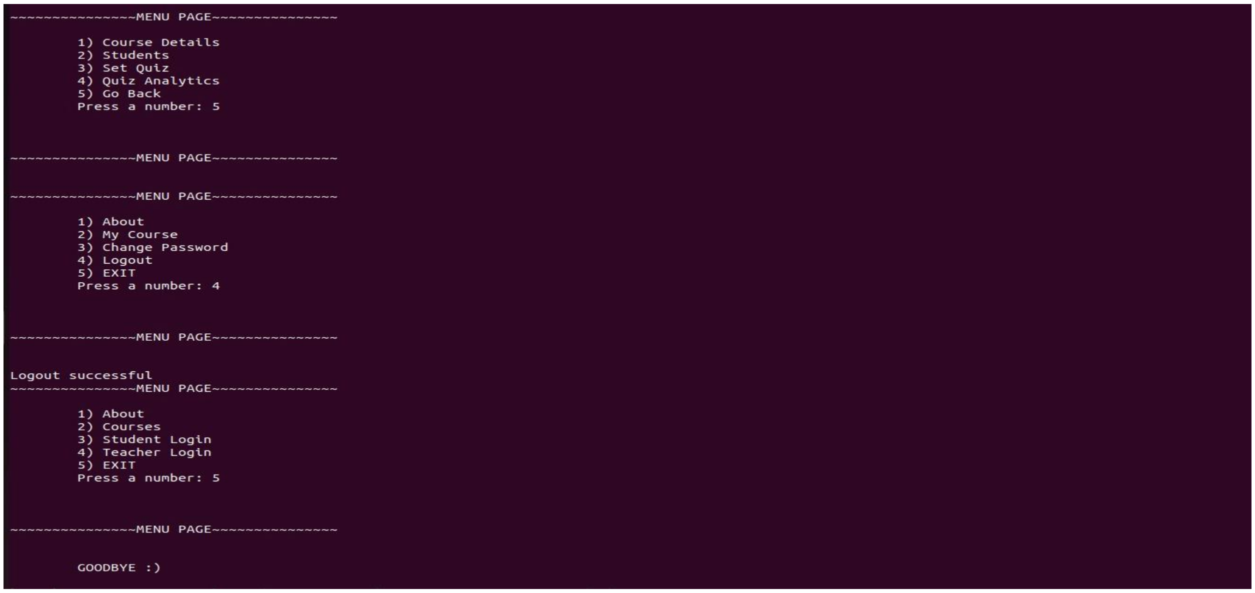
```
~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) My Courses
        3) Pending Quizzes
        4) Marks
        5) Change Password
        6) Logout
        7) EXIT
        Press a number: 4



~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~



        Course: Programming Fundamentals
        Topic: Variables and Data Types
        Marks: 5.6

~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) My Courses
        3) Pending Quizzes
        4) Marks
        5) Change Password
        6) Logout
        7) EXIT
        Press a number: 6



~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


Logout successful
```

In the same GenerateResult function, another CSV file is generated named QuizAnalytics, where the marks of each students according to the question number are stored. And when a teacher selects the option “Quiz Analytics,” the main menu function calls another member function of the Display class, where the data from the QuizAnalytics.csv is read and used to create a bar graph which shows the performance of the students in each Quiz.



```
~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) Course Details
        2) Students
        3) Set Quiz
        4) Quiz Analytics
        5) Go Back
        Press a number: 4



~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~



        The horizontal lines shows the number of students to score full marks
        Course: Programming Fundamentals        Topic: Variables and Data Types

        Number of Questions
        |
   Qs1  |-----------------------  23
   Qs2  |-------------------------  25
   Qs3  |------------------------------  30
   Qs4  |-------------------------  25
   Qs5  |-------------------  19
   Qs6  |---------------------------  27
   Qs7  |-----------------------  23
   Qs8  |-----------------------------  29
   Qs9  |--------------------  20
   Qs10 |---------------------  21
   Qs11 |-------------------  19
   Qs12 |---------------------------  27
        |_________________________________________________
                                                Num Students (50)
```

After that, the user can select the logout option, where the general main menu is redirected, and from there when pressing the EXIT option, the menu function returns to the main function and the program ends.



```
~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) Course Details
        2) Students
        3) Set Quiz
        4) Quiz Analytics
        5) Go Back
        Press a number: 5


~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) My Course
        3) Change Password
        4) Logout
        5) EXIT
        Press a number: 4


~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


Logout successful
~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~

        1) About
        2) Courses
        3) Student Login
        4) Teacher Login
        5) EXIT
        Press a number: 5


~~~~~~~~~~~~~~~MENU PAGE~~~~~~~~~~~~~~~


        GOODBYE :)
```

Conclusion:

In conclusion, the Examination System for Course Evaluations project has provided a robust and efficient solution for managing course evaluations. Through the utilization of object-oriented programming principles and the implementation of a well-structured architecture, the system offers a user-friendly interface for both students and teachers. The successful handling of CSV files, the implementation of class hierarchies, and the integration of key features such as quiz setting and analytics contribute to the project's success.