SEMESTER PROJECT

FINAL REPORT

Course : Object Oriented Programming

Project Title : SIBA Transcript Generator

Group Members:

Name	**CMS - ID**
Umair Ali	023-24-0106
Muhammad Hazique	023-24-0098
Ansa Arain	023-24-0078

Outline:

- Idea Clarity
- Problem Justification
- Methodology
- Implementation details
- Project Structure
- Screenshots

IDEA CLARITY:

The SIBA Transcript Generator is a desktop application developed in Java to automate the creation of academic transcripts for students at Sukkur IBA University. The system allows users to input student details (name, CMS-ID, department, semester, and transcript type: grades or marks), select or enter grades/marks for subjects, and generate a formatted transcript displaying grades, credit hours, and cumulative GPA. The application uses a multi-panel GUI built with Java Swing, ensuring an intuitive user experience. It supports various departments (e.g., Computer Science, Software Engineering, Accounting and Finance) and calculates grades and GPA based on predefined rules. The system adheres to Object-Oriented Programming (OOP) principles, such as encapsulation and Inheritance, to create a maintainable and user-friendly tool for academic administration.

PROBLEM JUSTIFICATION:

Generating academic transcripts manually is a time-consuming and error-prone process, especially in educational institutions with diverse departments and curricula. Administrators often struggle with calculating grades, GPAs, and formatting transcripts accurately, particularly when handling varying credit hours and grading schemes across semesters. The SIBA Transcript Generator addresses these challenges by providing an automated, user-friendly solution. It streamlines data entry, validates inputs, and produces printable transcripts with consistent formatting. By supporting multiple departments and semesters, the system ensures flexibility and reduces manual effort, making it a practical tool for academic staff to manage student records efficiently.

METHODOLOGY:

The development of the SIBA Transcript Generator followed a systematic approach to meet functional and academic requirements. The methodology included:

- **Requirement Analysis:**
  Our team identified key features through brainstorming, including student data input, grade/mark entry, GPA calculation, and transcript printing. The system needed to support multiple departments and flexible transcript types (grades or marks).

- **System Design:**
  The application was designed using OOP principles. The Info class encapsulates student data, while separate interface classes (FirstInterface, SecondInterface etc.) handle distinct GUI stages. Custom components (Box, Label etc.) ensure consistent UI design.

- **Technology Selection:**
  Java was chosen for its robust OOP support and cross-platform compatibility. Java Swing was used for the GUI due to its simplicity and flexibility. Static data from GetDepartments replaced the need for a database.

- **Implementation:**
  Development was iterative, starting with core classes (Info, GetDepartments), followed by GUI panels and logic for grade/GPA calculations. The print functionality was integrated in last.

- **Testing and Refinement:**
  Each panel was tested for functionality (e.g., input validation, navigation). Edge cases, such as invalid marks or empty fields, were handled with error messages. User feedback was incorporated to improve usability.

IMPLEMENTATION DETAILS

- **Class Design:**
  The Info class stores student data (e.g., name, CMS-ID, grades, GPA). The GetDepartments class provides department-specific subject lists, credit hours, and total marks. Custom UI classes (Box, Button, DropDownLists, etc.) encapsulate reusable components.

- **GUI Implementation:**
  - FirstInterface: Displays the main screen with a "Generate" button and SIBA logo.
  - SecondInterface: Collects student details (name, CMS-ID, department, semester, transcript type).
  - ThirdInterface: Allows grade selection or mark entry for subjects, with input validation.
  - FourthInterface: Displays the final transcript and supports printing.

- **Logic Implementation:**
  The FirstFrame class manages panel transitions and implements grade/GPA calculations. The getSubGrade method converts marks to grades (e.g., 93 = A), and getTotalGPA method computes the cumulative GPA, factoring in previous semester GPA if applicable.

- **Key Features:**
  - Input student details and select department/semester.
  - Support for grade-based or mark-based transcript generation.
  - Dynamic subject loading based on department and semester.
  - Input validation to prevent errors (e.g., invalid marks, empty fields).
  - Printable transcript with formatted student data, grades, and GPA.

- **Challenges:**
  Ensuring accurate GPA calculations across semesters, handling dynamic subject lists, and implementing print functionality were challenging (Got helped from sources i.e. Gemini, Grok_ai).

PROJECT STRUCTURE:

The project is organized into a modular structure to ensure clarity and maintainability. Below is the directory structure:

- **Project/:** Contains folders (source code, Sample Transcripts) and files (Final Report, Project Guidelines).

- **Source code/:** Contains all Java source files and compiled class files with a folder of images.
  - **Main.java:** Entry point, initializes FirstFrame.
  - **FirstFrame.java:** Main class, manages GUI panels and core logic.
  - **GetDepartments.java:** Provides department-specific subject, credit, and mark data.
  - **Interface Classes:** i.e. FirstInterface, SecondInterface, ThirdInterface, FourthInterface.
  - **Components Classes:** i.e. Box, Button, DropDownLists, Input, Label, Logo, MainButtons, MainLogo.
  - **Images/:** Stores image assets (e.g., IBALogo.png, GenerateIcon.png).

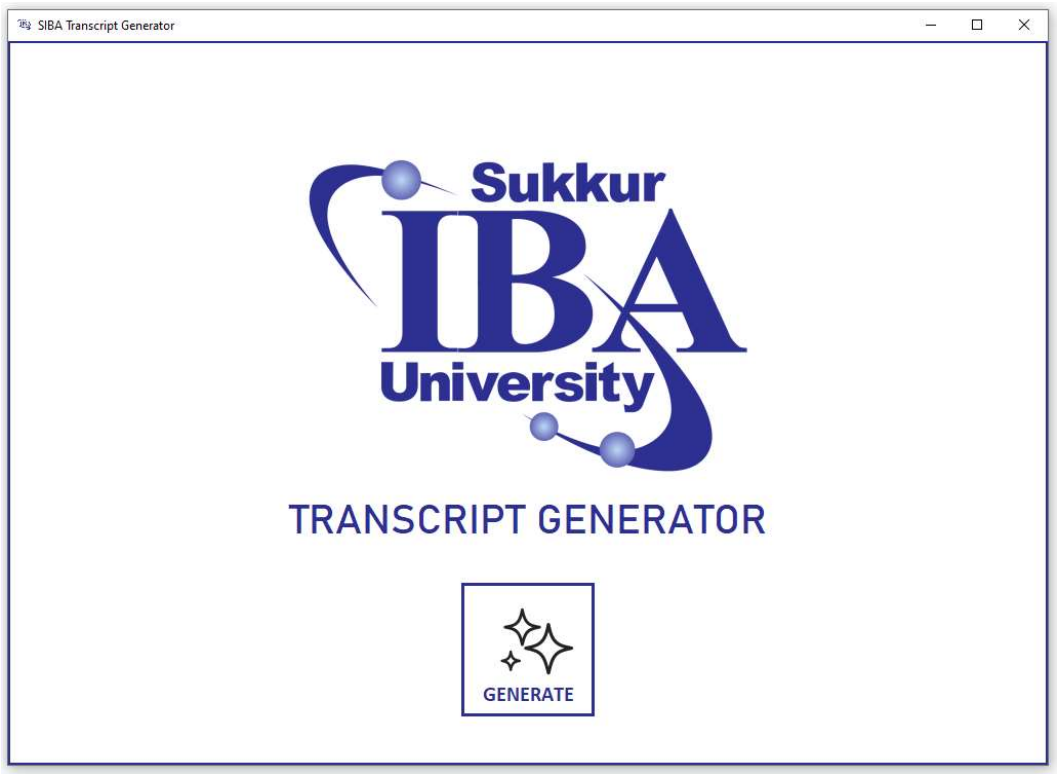
SCREENSHOTS

Execution commands:

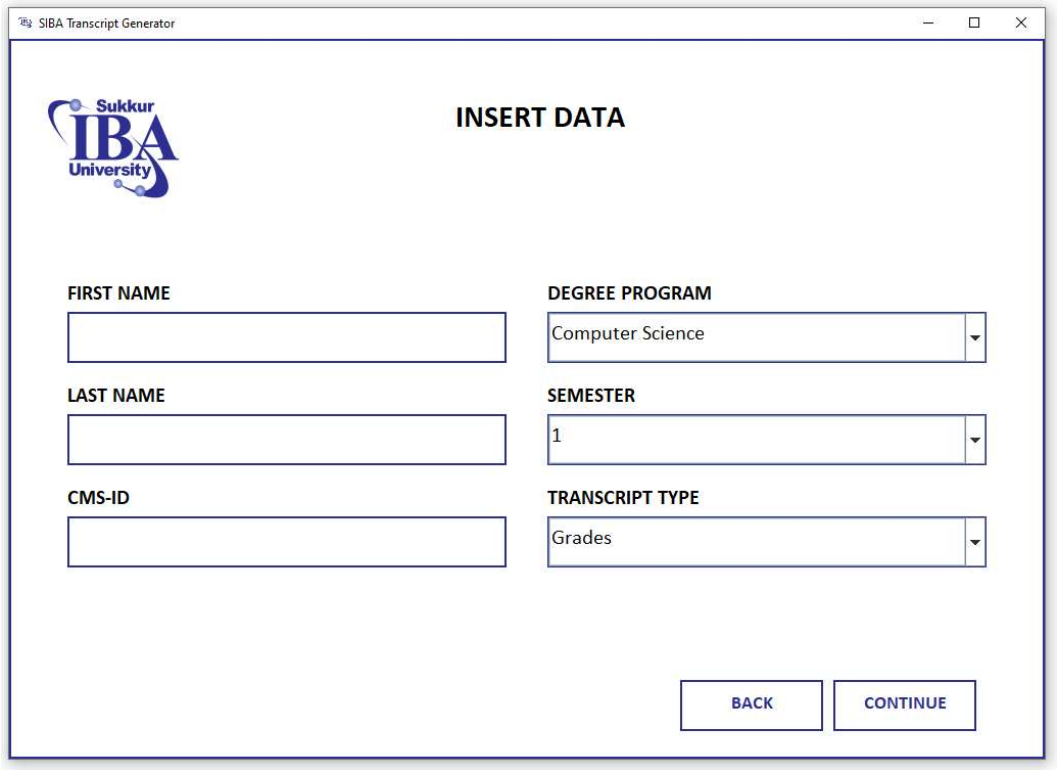
```
D:\University\CS Second Semester\Object Oriented Programming\Project\Source Code> javac Main.java

D:\University\CS Second Semester\Object Oriented Programming\Project\Source Code> java Main
```

Overall Interfaces:

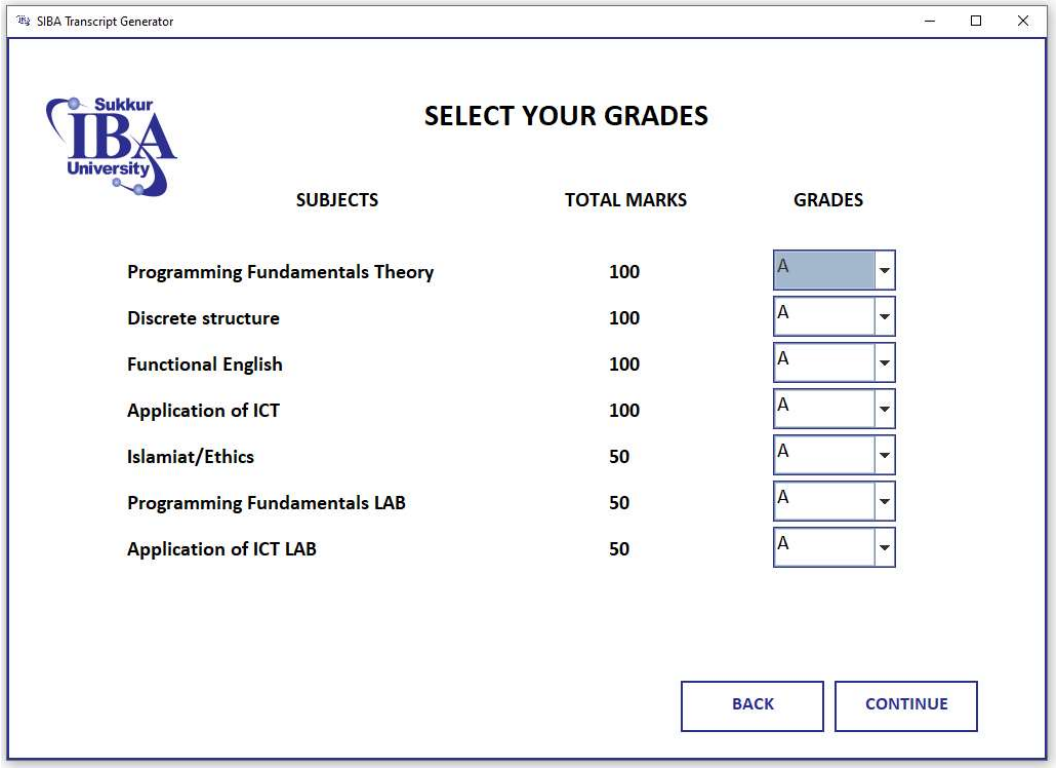
- **First Interface**



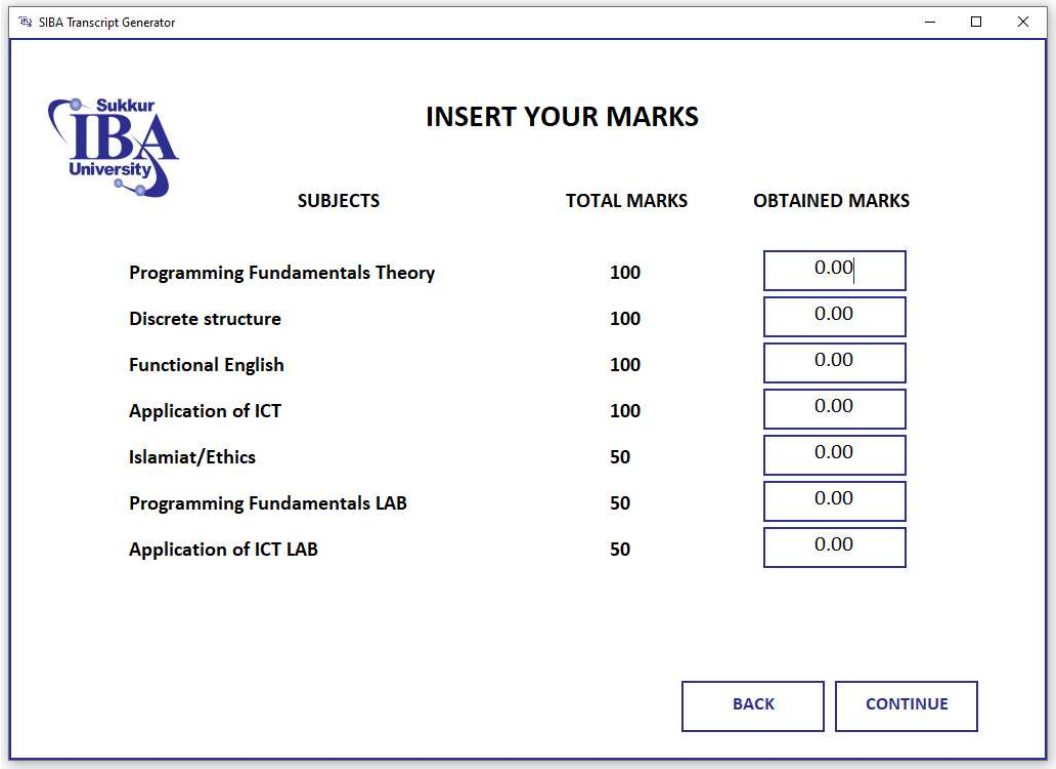
- **Second Interface (when user pressed the generate button)**



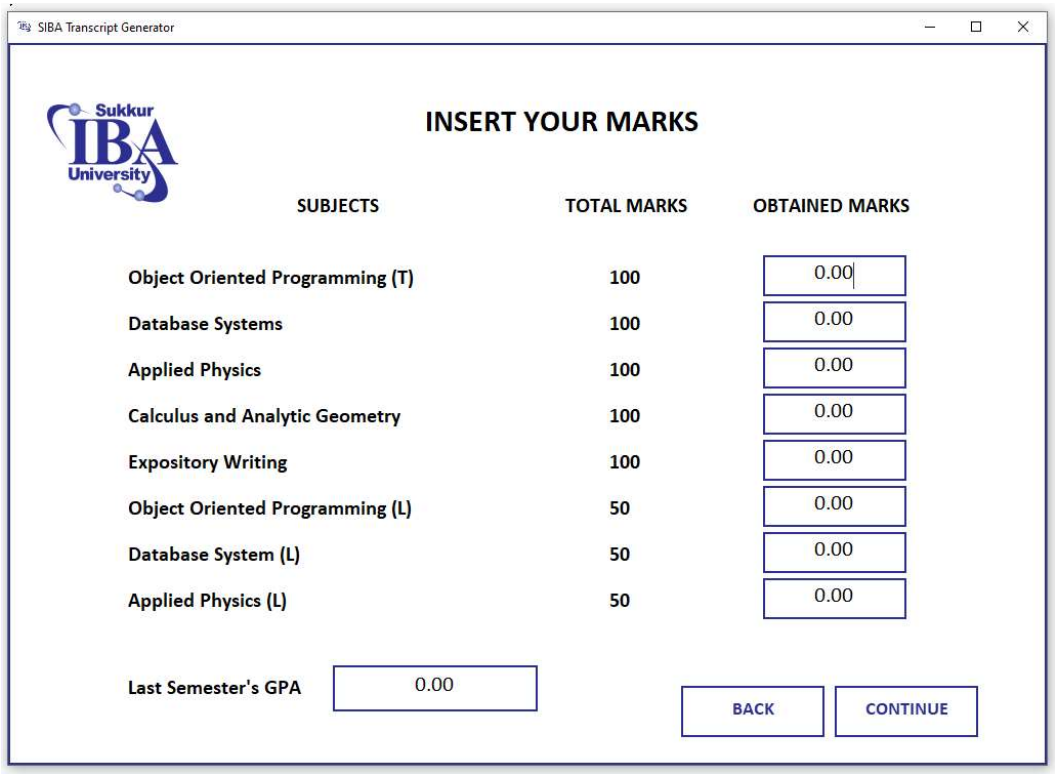
- **Third Interface (when user selected grades in type of transcript)**



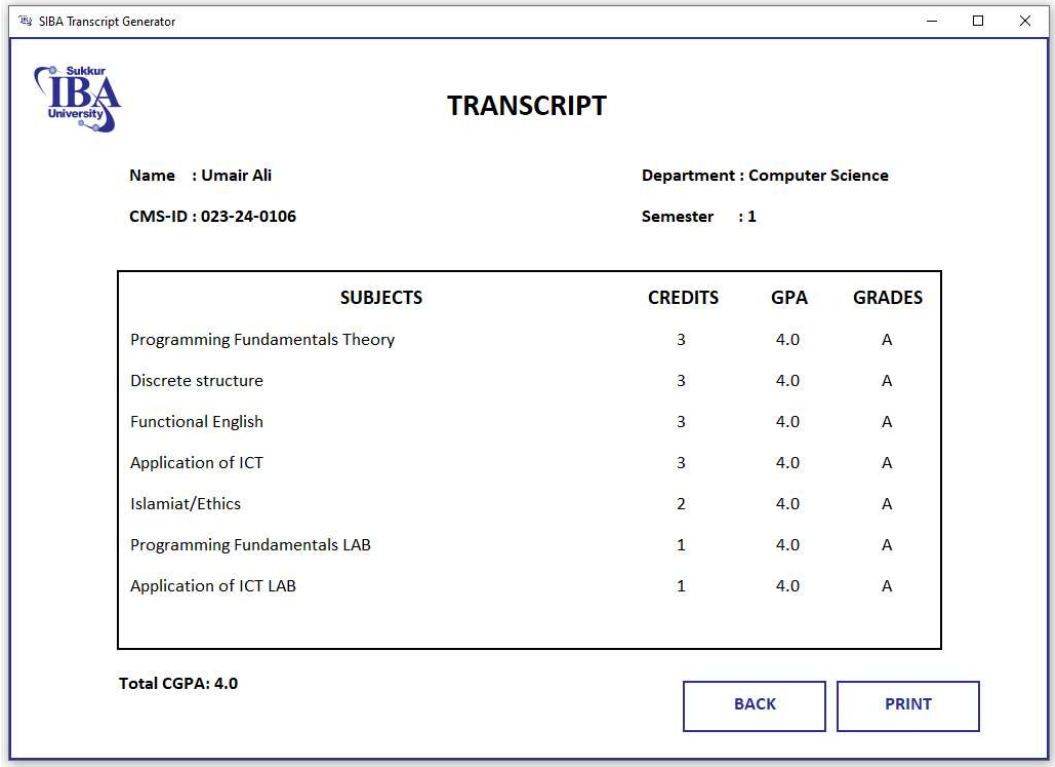
- **Third Interface (when user selected marks in type of transcript)**



- **Third Interface (when user selected marks in type of transcript and semester other than 1)**

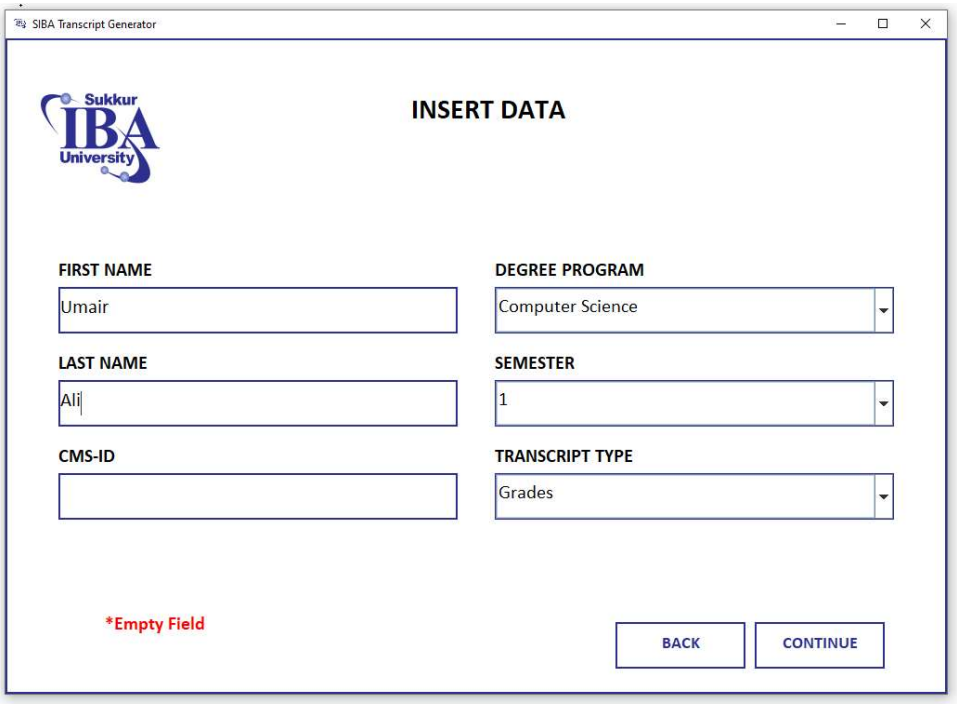


- **Fourth Interface**

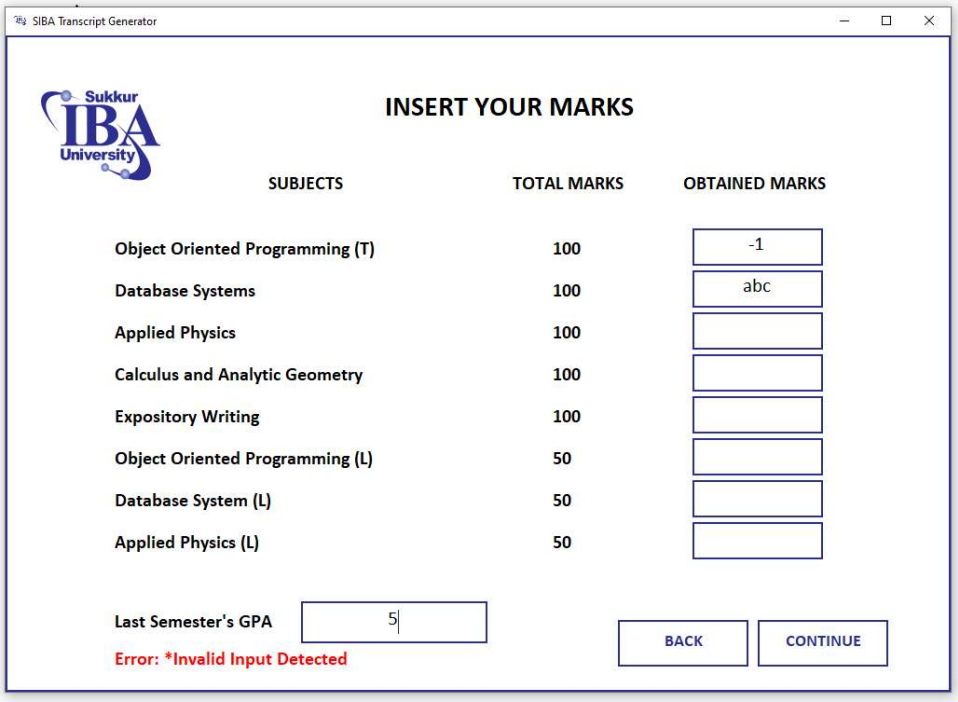


Invalid Input Reaction:

- **If User leaves any field blank the program would not proceed**

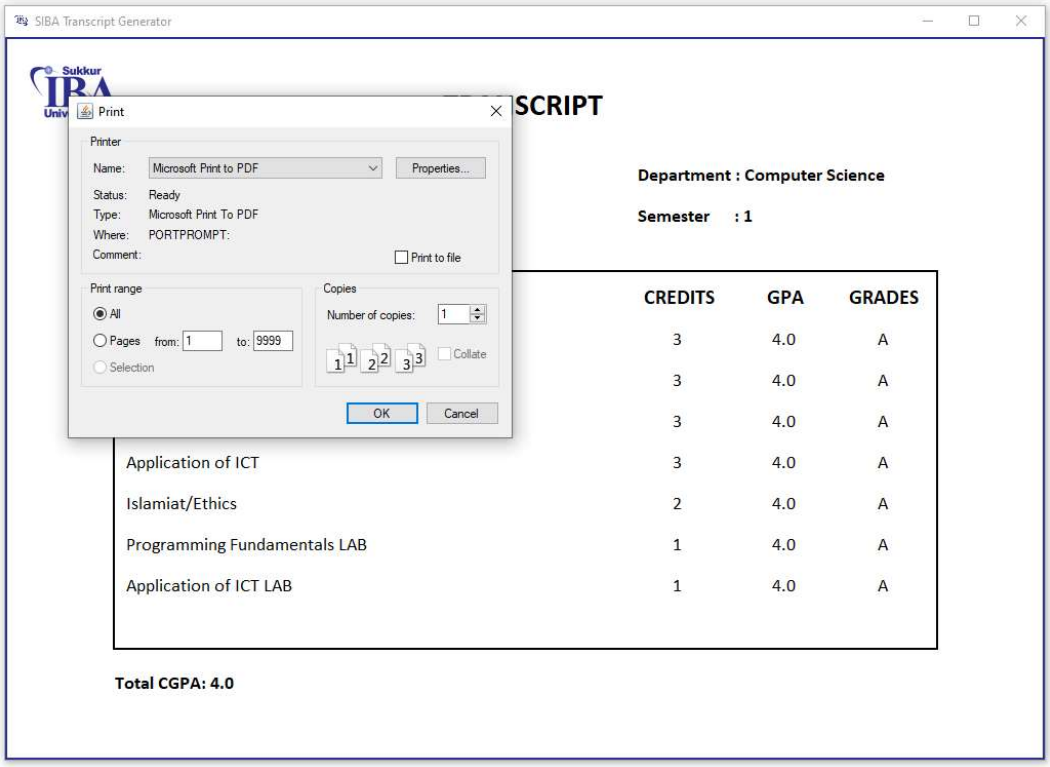


- **If User inputs the value that is not allowed, the program would not proceed**
  - If user inserted marks less than 0 and greater than total marks of corresponding subject
  - If user inserted text or non-numerical value in any field
  - If user inserted his/her last semester GPA not between 0 to 4

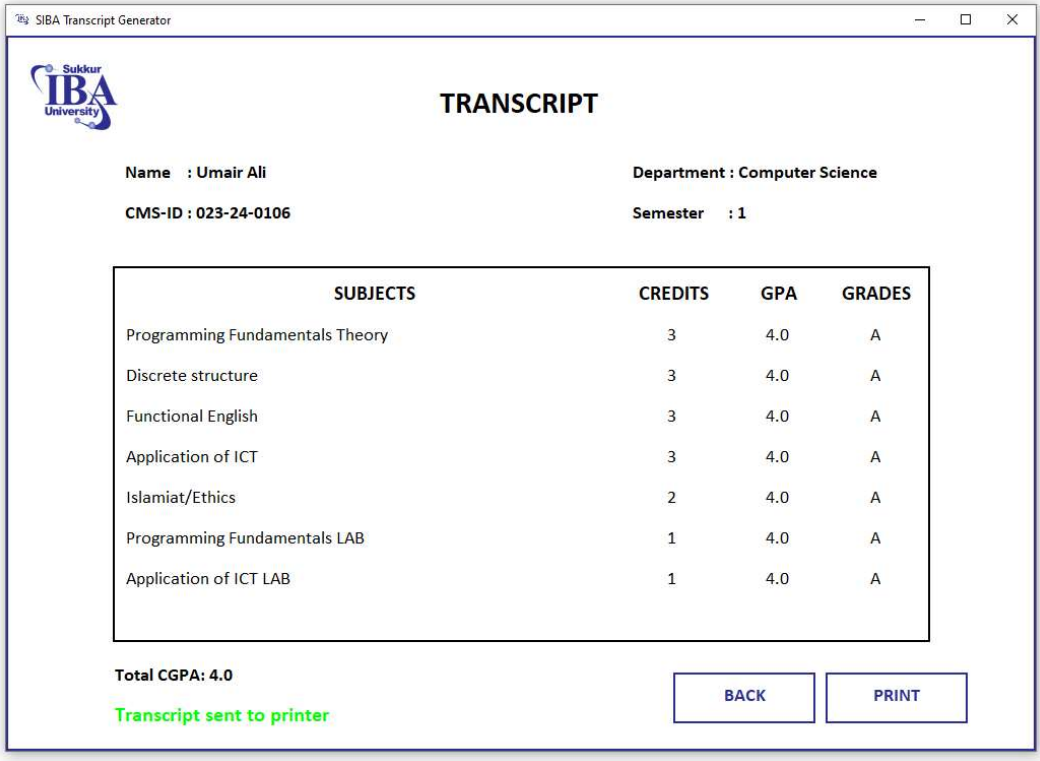


Print Functionality:

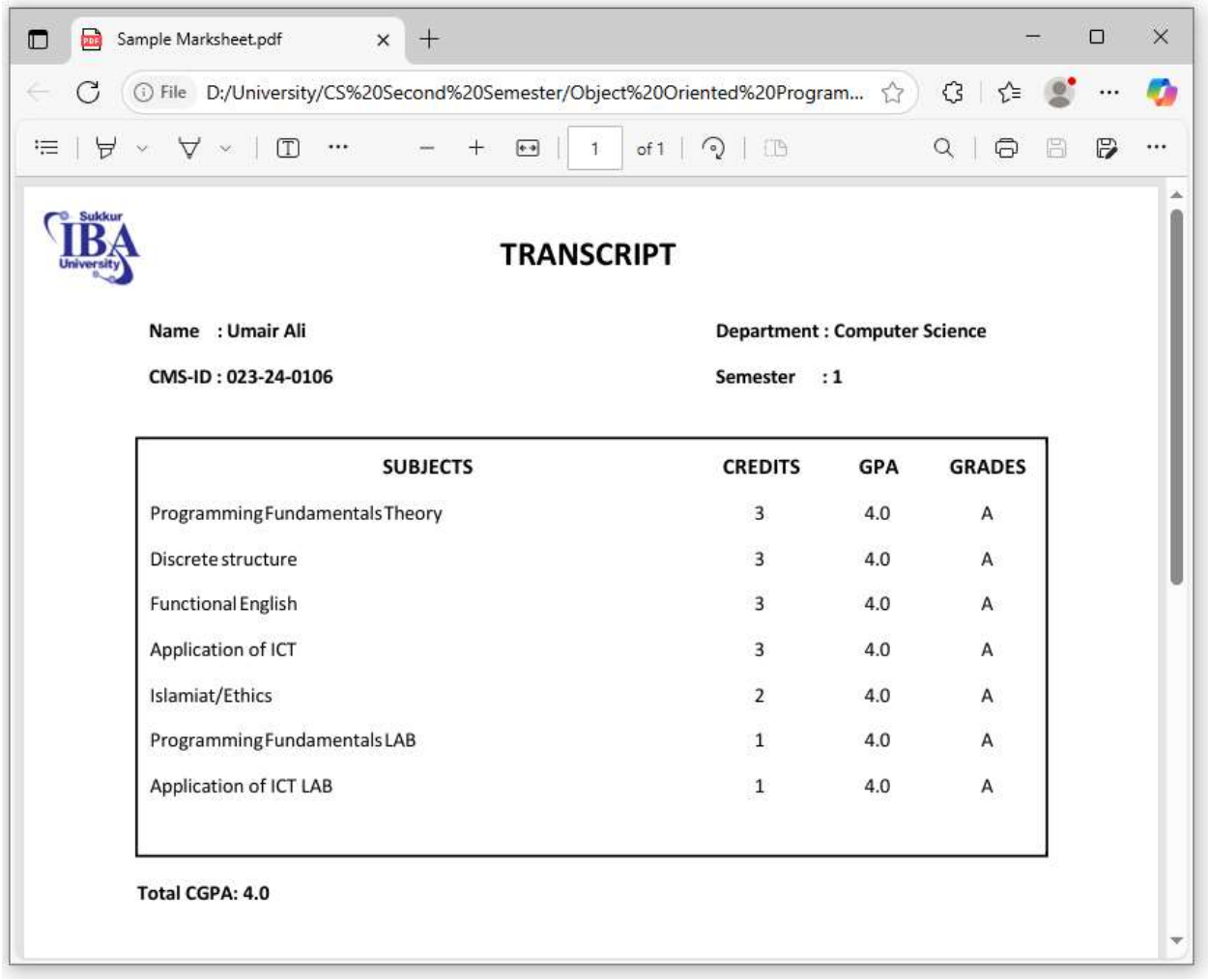
- **If User pressed the print button, the print dialogue box will appear**
  - The buttons would disappear until the print dialogue box is not closed



- **After successful printing**



Result Transcript:



Thank you

The end