# Worksheet 2

**-Utsab Poudel**

**Tasks**

**Task 1: Basic student grading system prototype using classes and objects. [30 Marks]**

Write a program that manages a simple student grade calculator with the following requirements. Create a Student class that has:

1. Student name (string)
2. Three subject marks (integers)
3. A basic member function to calculate average

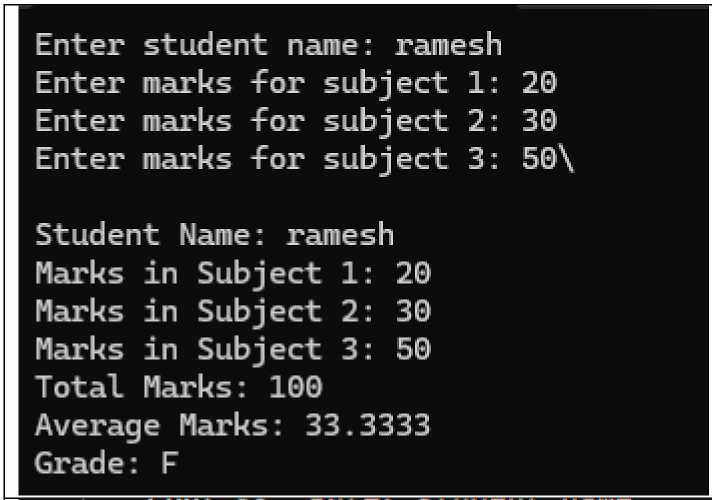
The program should:

1. Accept student details (name and marks) from user input
2. Calculate and display:
    1. Total marks
    2. Average marks
    3. Grade (A for ≥90%, B for ≥80%, C for ≥70%, D for ≥60%, F for <60%)
3. Display a message if any mark is below 0 or above 100

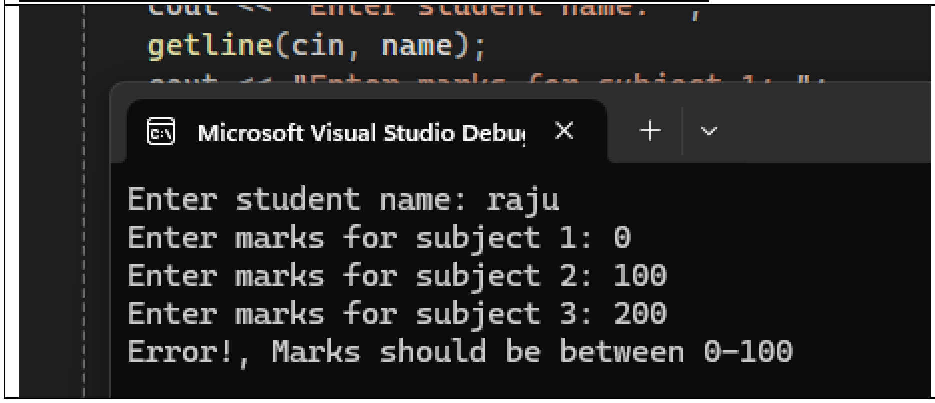
```
#include <iostream>
#include <string>
using namespace std;

class Student {
private:
    string name;
    int marks1, marks2, marks3;

public:
    void getDetails() {
        cout << "Enter student name: ";
        getline(cin, name);
        cout << "Enter marks for subject 1: ";
        cin >> marks1;
        cout << "Enter marks for subject 2: ";
        cin >> marks2;
        cout << "Enter marks for subject 3: ";
        cin >> marks3;
```

```cpp
        if (marks1 < 0 || marks1 > 100 || marks2 < 0 || marks2 > 100 || marks3 < 0
|| marks3 > 100) {
                   int a = 1; // Dummy variable to throw an exception
            throw(a);
        }
    }

    void display() {
        cout << "\nStudent Name: " << name << endl;
        cout << "Marks in Subject 1: " << marks1 << endl;
        cout << "Marks in Subject 2: " << marks2 << endl;
        cout << "Marks in Subject 3: " << marks3 << endl;
    }

    void average() {
        int total = marks1 + marks2 + marks3;
        float avg = total / 3.0;
        cout << "Total Marks: " << total << endl;
        cout << "Average Marks: " << avg << endl;

        cout << "Grade: ";
        if (avg >= 90) {
            cout << "A" << endl;
        }
        else if (avg >= 80) {
            cout << "B" << endl;
        }
        else if (avg >= 70) {
            cout << "C" << endl;
        }
        else if (avg >= 60) {
            cout << "D" << endl;
        }
        else {
            cout << "F" << endl;
        }
    }
};

int main() {
    Student s1;
    try {
        s1.getDetails();
        s1.display();
        s1.average();
    }
    catch (int a) {
        cout << "Error!, Marks should be between 0-100" << endl;
    }

    return 0;
}
```

Output:



```
Enter student name: ramesh
Enter marks for subject 1: 20
Enter marks for subject 2: 30
Enter marks for subject 3: 50\

Student Name: ramesh
Marks in Subject 1: 20
Marks in Subject 2: 30
Marks in Subject 3: 50
Total Marks: 100
Average Marks: 33.3333
Grade: F
```



```
    cout << "Enter student name: ";
    getline(cin, name);
    cout << "Enter marks for subject 1: ";
```

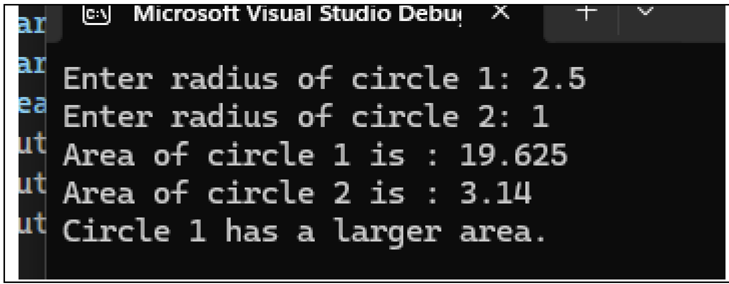
```
Enter student name: raju
Enter marks for subject 1: 0
Enter marks for subject 2: 100
Enter marks for subject 3: 200
Error!, Marks should be between 0-100
```

**Task 2: Programming assignments: All questions are mandatory**

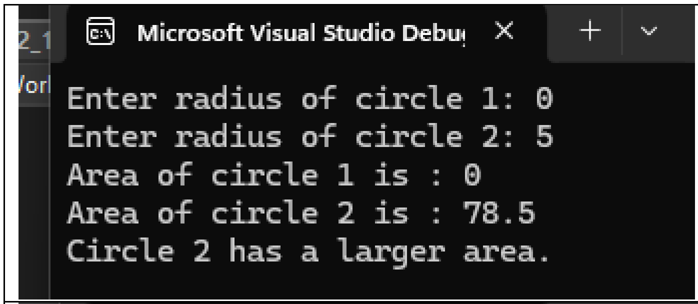
1. Write a program with a class Circle having:
    1. Private member: radius (float)
    2. A constructor to initialize radius
    3. A friend function compareTwoCircles that takes two Circle objects and prints which circle has the larger area

```
#include <iostream>
using namespace std;
class Circle {
private:
	float radius;
public:
        Circle(float r) {
        radius = r;
        }

        friend void compareTwoCircles(Circle c1, Circle c2);
};
void compareTwoCircles(Circle c1, Circle c2) {
        float PI = 3.14;
        float area1 = PI * c1.radius * c1.radius;
        float area2 = PI * c2.radius * c2.radius;
        if (area1 > area2) {
                cout << "Area of circle 1 is : " << area1 << endl;
                cout << "Area of circle 2 is : " << area2 << endl;
                cout << "Circle 1 has a larger area." << endl;
        }
        else if (area1 < area2) {
                cout << "Area of circle 1 is : " << area1 << endl;
                cout << "Area of circle 2 is : " << area2 << endl;
                cout << "Circle 2 has a larger area." << endl;
        }
        else {
                cout << "Area of circle 1 is : " << area1 << endl;
                cout << "Area of circle 2 is : " << area2 << endl;
                cout << "Both circles have the same area." << endl;
        }
}
int main() {
        float a, b;
        cout << "Enter radius of circle 1: ";
        cin >> a;
        cout << "Enter radius of circle 2: ";
        cin >> b;
        Circle cir1(a);
        Circle cir2(b);
        compareTwoCircles(cir1, cir2);
        return 0;
}
```

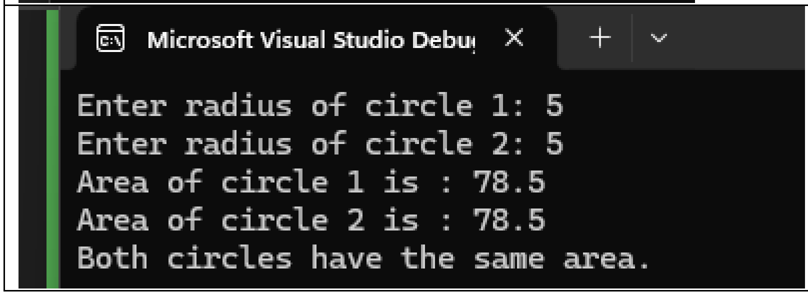
**Output:**



```
Enter radius of circle 1: 2.5
Enter radius of circle 2: 1
Area of circle 1 is : 19.625
Area of circle 2 is : 3.14
Circle 1 has a larger area.
```



```
Enter radius of circle 1: 0
Enter radius of circle 2: 5
Area of circle 1 is : 0
Area of circle 2 is : 78.5
Circle 2 has a larger area.
```



```
Enter radius of circle 1: 5
Enter radius of circle 2: 5
Area of circle 1 is : 78.5
Area of circle 2 is : 78.5
Both circles have the same area.
```

2. Create a program with these overloaded functions named findMax:

   1. One that finds maximum between two integers

   2. One that finds maximum between two floating-point numbers

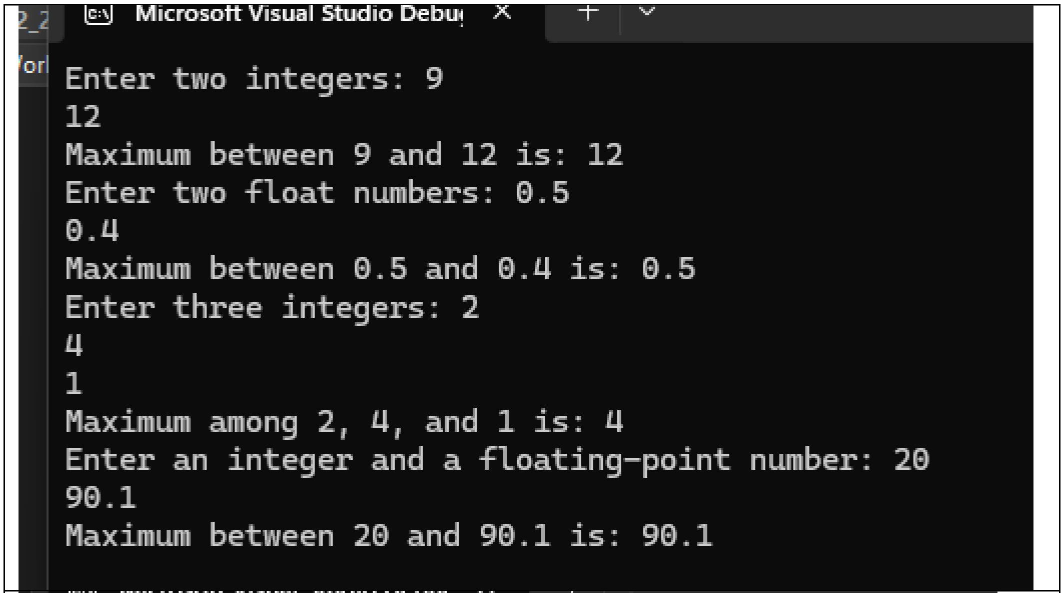
   3. One that finds maximum among three integers

   4. One that finds maximum between an integer and a float

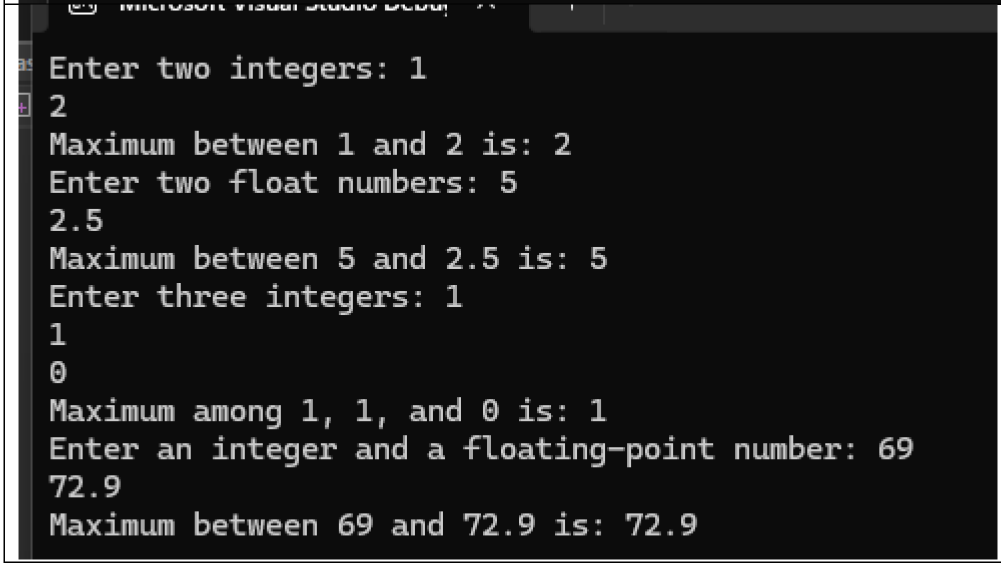
**[50 marks]**

```
#include <iostream>
using namespace std;

class Maximum {
private:
    int maxInt;
    float maxFloat;

public:
    // two integers
    void Max(int a, int b) {
        if (a > b) {
            maxInt = a;
        }
        else {
            maxInt = b;
        }
        cout << "Maximum between " << a << " and " << b << " is: " <<
maxInt << endl;
    }
    // two floats
    void Max(float a, float b) {
        if (a > b) {
            maxFloat = a;
        }
        else {
            maxFloat = b;
        }
        cout << "Maximum between " << a << " and " << b << " is: " <<
maxFloat << endl;
    }

    // three ints
    void Max(int a, int b, int c) {
        if (a > b) {
            maxInt = a;
        }
        else {
            maxInt = b;
        }

        if (c > maxInt) {
            maxInt = c;
        }

        cout << "Maximum among " << a << ", " << b << ", and " << c
<< " is: " << maxInt << endl;
    }

    // an int and a float
    void Max(int a, float b) {
        if (a > b) {
            maxFloat = a;
        }
        else {
            maxFloat = b;
        }
        cout << "Maximum between " << a << " and " << b << " is: " <<
maxFloat << endl;
    }

    void run() {
        int int1, int2, int3;
        float float1, float2;

        cout << "Enter two integers: ";
        cin >> int1 >> int2;
        Max(int1, int2);

        cout << "Enter two float numbers: ";
        cin >> float1 >> float2;
        Max(float1, float2);

        cout << "Enter three integers: ";
        cin >> int1 >> int2 >> int3;
        Max(int1, int2, int3);

        cout << "Enter an integer and a floating-point number: ";
        cin >> int1 >> float1;
        Max(int1, float1);
    }
};

int main() {
    Maximum m1;
    m1.run();
    return 0;
}
```

**Output:**



```
Enter two integers: 9
12
Maximum between 9 and 12 is: 12
Enter two float numbers: 0.5
0.4
Maximum between 0.5 and 0.4 is: 0.5
Enter three integers: 2
4
1
Maximum among 2, 4, and 1 is: 4
Enter an integer and a floating-point number: 20
90.1
Maximum between 20 and 90.1 is: 90.1
```



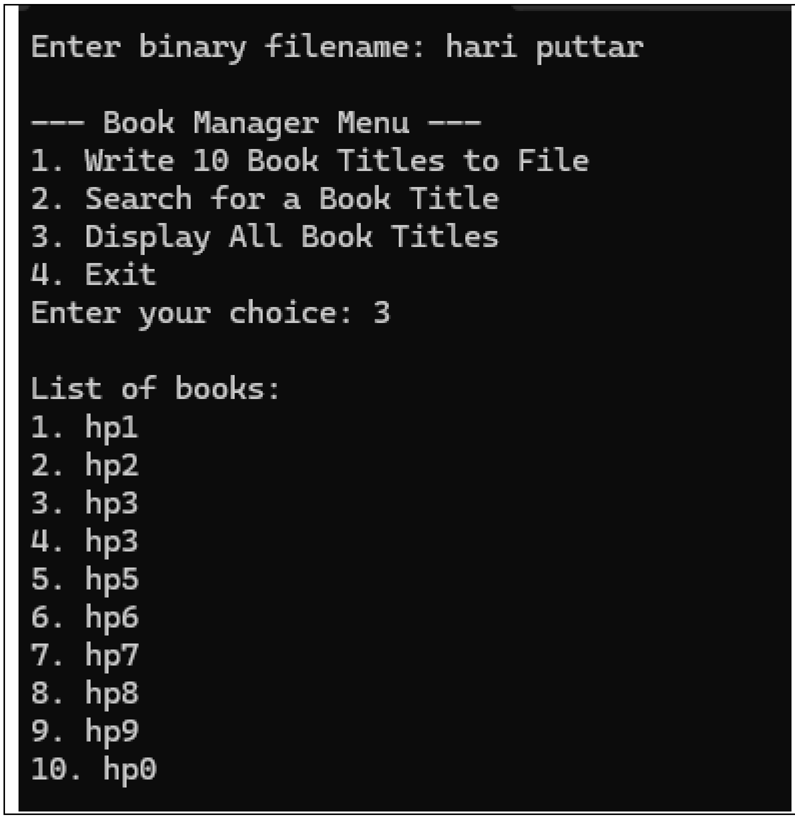
```
Enter two integers: 1
2
Maximum between 1 and 2 is: 2
Enter two float numbers: 5
2.5
Maximum between 5 and 2.5 is: 5
Enter three integers: 1
1
0
Maximum among 1, 1, and 0 is: 1
Enter an integer and a floating-point number: 69
72.9
Maximum between 69 and 72.9 is: 72.9
```

## Task 3: Basics of File Handling

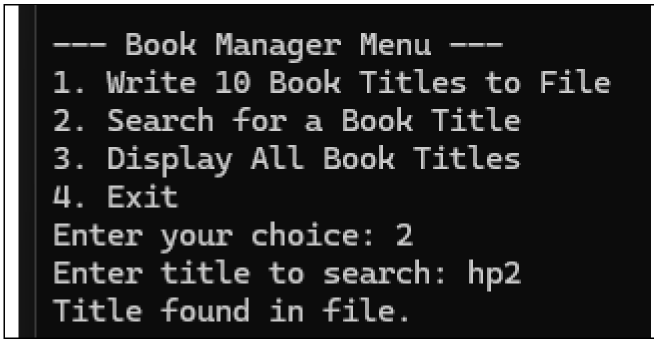
Write a program that reads the titles of 10 books (use an array of 150 characters) and writes them in a binary file selected by the user. The program should read a title and display a message to indicate if it is contained in the file or not.

```
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;

class BookManager {
    char filename[100];

public:
    void getFilename() {
        cout << "Enter binary filename: ";
        cin.getline(filename, 100);
    }

    void writeBooks() {
        ofstream file(filename, ios::binary);
        if (!file) {
            cout << "Error opening file.\n";
            return;
        }

        char title[150];
        for (int i = 0; i < 10; i++) {
            cout << "Enter title " << (i + 1) << ": ";
            //function is used to set all bytes of the title array to zero (or the
null character '\0')
            memset(title, 0, 150);
            cin.getline(title, 150);
            file.write(title, 150);
        }

        file.close();
        cout << "Books written to binary file.\n";
    }

    void searchBook() {
        ifstream file(filename, ios::binary);
        if (!file) {
            cout << "File not found.\n";
            return;
        }

        char search[150];
        cout << "Enter title to search: ";
        cin.getline(search, 150);

        char title[150];
        bool found = false;

        while (file.read(title, 150)) {
            // function in C/C++ is used to compare a specified number of
characters between two string
            if (strncmp(title, search, 150) == 0) {
                found = true;
                break;
            }
        }

        if (found) {
            cout << "Title found in file.\n";
        }
        else {
            cout << "Title not found.\n";
        }

        file.close();
    }

    void displayBooks() {
        ifstream file(filename, ios::binary);
        if (!file) {
            cout << "File not found.\n";
            return;
        }

        char title[150];
        int count = 1;

        cout << "\nList of books:\n";
        while (file.read(title, 150)) {
            cout << count << ". " << title << "\n";
            count++;
        }

        file.close();
    }

    void menu() {
        getFilename();

        int choice;
        do {
            cout << "\n--- Book Manager Menu ---\n";
            cout << "1. Write 10 Book Titles to File\n";
            cout << "2. Search for a Book Title\n";
            cout << "3. Display All Book Titles\n";
            cout << "4. Exit\n";
            cout << "Enter your choice: ";
            cin >> choice;
            cin.ignore(); // To clear leftover newline after cin , to prevent
issues with input

            switch (choice) {
            case 1:
                writeBooks();
                break;
            case 2:
                searchBook();
                break;
            case 3:
                displayBooks();
                break;
            case 4:
                cout << "Exiting program.\n";
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
            }
        } while (choice != 4);
    }
};

int main() {
    BookManager m1;
    m1.menu();
    return 0;
}
```

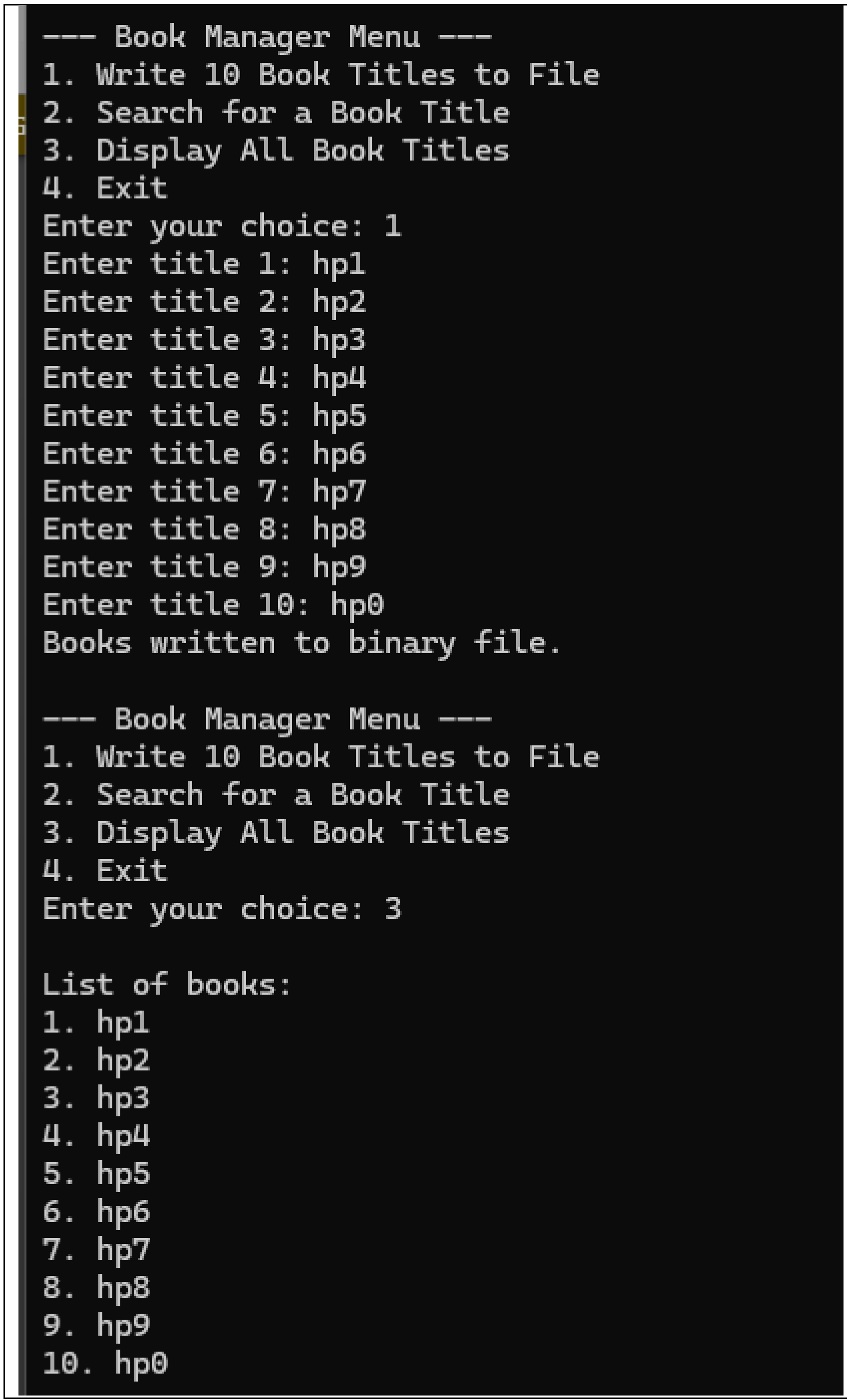
**Output:**



```
Enter binary filename: hari puttar

--- Book Manager Menu ---
1. Write 10 Book Titles to File
2. Search for a Book Title
3. Display All Book Titles
4. Exit
Enter your choice: 3

List of books:
1. hp1
2. hp2
3. hp3
4. hp3
5. hp5
6. hp6
7. hp7
8. hp8
9. hp9
10. hp0
```



```
--- Book Manager Menu ---
1. Write 10 Book Titles to File
2. Search for a Book Title
3. Display All Book Titles
4. Exit
Enter your choice: 2
Enter title to search: hp2
Title found in file.
```



```
--- Book Manager Menu ---
1. Write 10 Book Titles to File
2. Search for a Book Title
3. Display All Book Titles
4. Exit
Enter your choice: 1
Enter title 1: hp1
Enter title 2: hp2
Enter title 3: hp3
Enter title 4: hp4
Enter title 5: hp5
Enter title 6: hp6
Enter title 7: hp7
Enter title 8: hp8
Enter title 9: hp9
Enter title 10: hp0
Books written to binary file.

--- Book Manager Menu ---
1. Write 10 Book Titles to File
2. Search for a Book Title
3. Display All Book Titles
4. Exit
Enter your choice: 3

List of books:
1. hp1
2. hp2
3. hp3
4. hp4
5. hp5
6. hp6
7. hp7
8. hp8
9. hp9
10. hp0
```

Create a program that:

1. Reads student records (roll, name, marks) from a text file
2. Throws an exception if marks are not between 0 and 100

3. Allows adding new records with proper validation

4. Saves modified records back to file

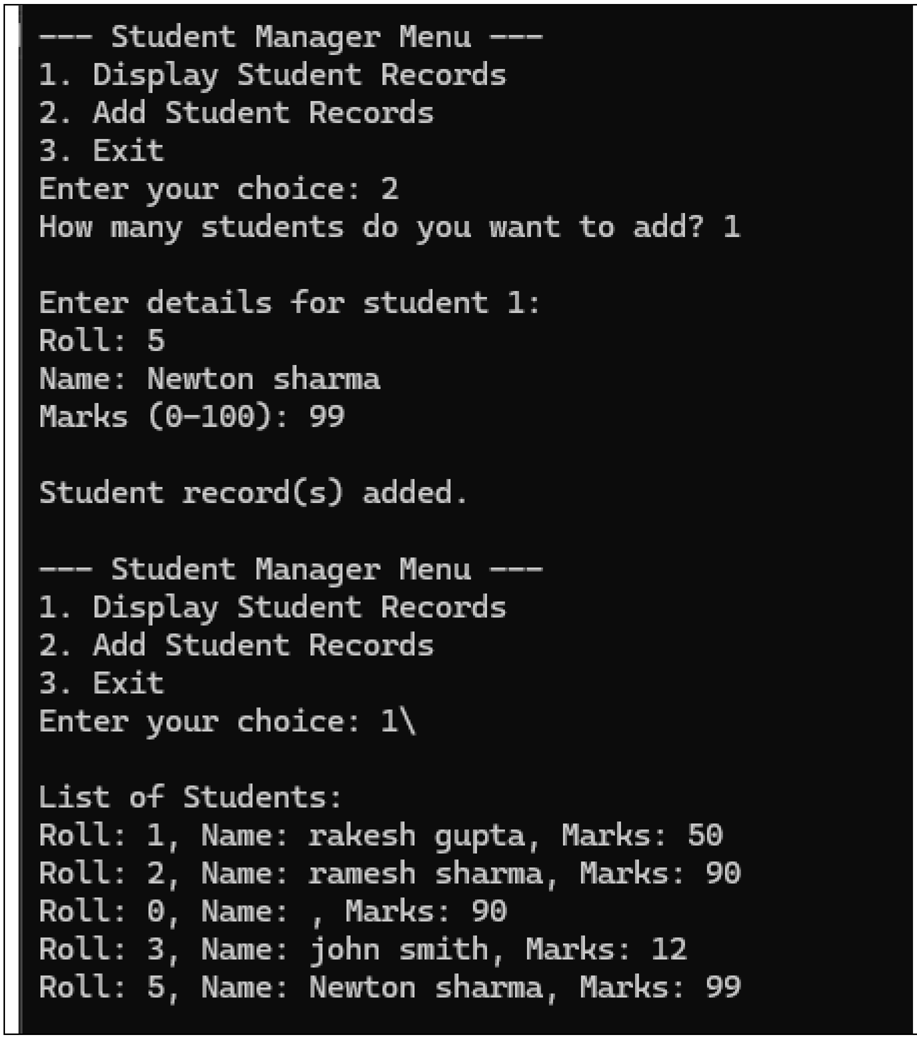
**[20 marks]**

**Note:**

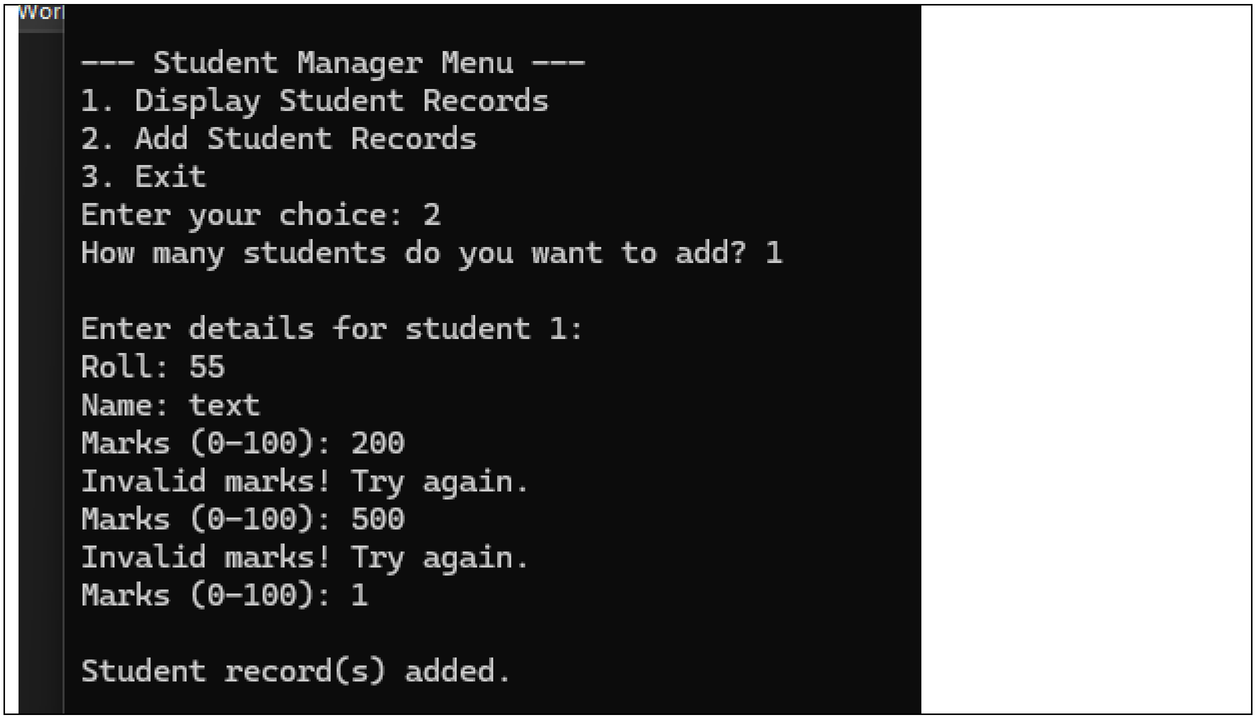
This is a simple code that allows us to read a file and display its content and save these in a txt file . The only limitation is we cannot update/delete

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

class StudentManager {
public:
    void displayStudents() {
        ifstream file("student.txt");
        if (!file) {
            cout << "File not found.\n";
            return;
        }

        int roll, marks;
        string name;

        cout << "\nList of Students:\n";
        while (file >> roll >> ws) {
            getline(file, name, '|');
            file >> marks;
            cout << "Roll: " << roll << ", Name: " << name << ", Marks: " << marks
<< "\n";
        }

        file.close();
    }

    void addStudents() {
        ofstream file("student.txt", ios::app);
        if (!file) {
            cout << "Error opening file.\n";
            return;
        }

        int n;
        cout << "How many students do you want to add? ";
        cin >> n;

        for (int i = 0; i < n; i++) {
            int roll, marks;
            string name;

            cout << "\nEnter details for student " << (i + 1) << ":\n";
            cout << "Roll: ";
            cin >> roll;

            cout << "Name: ";
                    // Using ws to consume any leading whitespace before getline
            cin >> ws;
            getline(cin, name);

            cout << "Marks (0-100): ";
            cin >> marks;

            if (marks >= 0 && marks <= 100) {
                file << roll << " " << name << "|" << marks << "\n";
            }
            else {
                cout << "Invalid marks. Skipping student.\n";
            }
        }

        file.close();
        cout << "\nStudent record(s) added.\n";
    }

    void menu() {
        int choice = 0;
        while (choice != 3) {
            cout << "\n--- Student Manager Menu ---\n";
            cout << "1. Display Student Records\n";
            cout << "2. Add Student Records\n";
            cout << "3. Exit\n";
            cout << "Enter your choice: ";
            cin >> choice;

            if (choice == 1) {
                displayStudents();
            }
            else if (choice == 2) {
                addStudents();
            }
            else if (choice == 3) {
                cout << "Exiting program.\n";
            }
            else {
                cout << "Invalid choice.\n";
            }
        }
    }
};

int main() {
    StudentManager m1;
    m1.menu();
    return 0;
}
```

**Output:**



```
--- Student Manager Menu ---
1. Display Student Records
2. Add Student Records
3. Exit
Enter your choice: 2
How many students do you want to add? 1

Enter details for student 1:
Roll: 5
Name: Newton sharma
Marks (0-100): 99

Student record(s) added.

--- Student Manager Menu ---
1. Display Student Records
2. Add Student Records
3. Exit
Enter your choice: 1\

List of Students:
Roll: 1, Name: rakesh gupta, Marks: 50
Roll: 2, Name: ramesh sharma, Marks: 90
Roll: 0, Name: , Marks: 90
Roll: 3, Name: john smith, Marks: 12
Roll: 5, Name: Newton sharma, Marks: 99
```



```
--- Student Manager Menu ---
1. Display Student Records
2. Add Student Records
3. Exit
Enter your choice: 2
How many students do you want to add? 1

Enter details for student 1:
Roll: 55
Name: text
Marks (0-100): 200
Invalid marks! Try again.
Marks (0-100): 500
Invalid marks! Try again.
Marks (0-100): 1

Student record(s) added.
```

**Task 4**

- Check and commit all your solutions.
- This task carries no marks but it is mandatory. Ensure that your solution is visible to us.