

This problem tests your knowledge of classes, composition, inheritance, polymorphism and file handling. Read the questions carefully and answer precisely.

(a)

Declare and implement a class called **TableOfContents**. It contains a dynamic array of type **Content**, defined by the following struct:

```
struct Content{
    string name; //chapter name
    int pno; //page number of the chapter
};
```

TableOfContents also contains an integer **n** for the number of contents. The class also has a constructor **TableOfContents(const string &fname)** which receives a file name and reads the contents from that file. It is a text file in which the first line contains the number of chapters and then alternate lines contain chapter name and corresponding page number. Following is an example file with only two contents.

```
2
Introduction to Computers
1
Introduction to C++ Programming
25
```

The class should implement a function called **printContents** which prints chapter names and the corresponding page numbers, line by line.

(b)

Read the following description carefully and implement a system of classes accordingly. For each class you must implement appropriate constructors and methods only. For the **Date** class you can simply use the following interface without implementing it.

```
class Date{
    int day,month,year;
public:
    Date(int,int,int);
    void SetDate(int,int,int);
    int operator – (const Date&); //difference between two dates in number of days
};
```

A Book *contains* a table of contents, a string ISBN (which is unique for each book) and a string title. A book is classified into two main categories i.e., a reference book, which cannot be issued, and any other book that is issuable. All books can print their table of contents but reference books print only the number of chapters and page numbers of the first and the last chapter, while issuable books actually print all the chapter

names and their corresponding page numbers, line by line.

A **dictionary** and an **encyclopedia** are both reference books, while a **textbook** and a **general book** are both issuable books.

Every reference book can print a short description of itself. The description of a dictionary is: "**title** is a list of

words and their meanings" and the description for an encyclopedia is: "**title** contains articles on various topics", where **title** is simply the title of the book.

Every issuable book contains a date of issuance and can calculate fine for a book by checking if the return date (passed as parameter) is more than fourteen days away (in case of general book) or more than seven days away (in case of text book) from the date of issuance. A **10 rupee** fine per day is charged for a text book and a **5 rupee** fine per day is charged for a general book. An issuable book stores the last value of calculated fine (which is set to zero in the constructor), for later use. Issuable books can be issued by setting the issuance date to a date passed as parameter.

The system only allows objects of the four types: **dictionary**, **encyclopedia**, **general book**, and **text book**, to be instantiated.

i)

Provide the number and name of the classes that are required to implement the above system.

ii)

Provide the declarations for each required class (Only declare the needed methods)

iii)

Provide Implementation for each declared class. For each class you must implement appropriate constructors and methods only.

(c)

Give the declaration of a class called **Library** which contains dynamic arrays of issuable and reference books, called **shelf1** and **shelf2** respectively, and the numbers of items in both. For this class implement the following member method only. (Note: You might assume that all the data is already stored in the library; you don't need to take any input):

printBookDescriptions(): this function goes through the array **shelf2** and prints the descriptions of all the reference books inside it.

```
#include <iostream>
#include <fstream>
#include <string>
```

```

using namespace std;

// (a)
struct Content {
    string name;    // chapter name
    int pno;        // page number of the chapter
};

class TableOfContents {
private:
    Content* contents;
    int n;
public:
    TableOfContents(const string& fname) : contents(nullptr),
n(0) {
        ifstream file(fname);
        if (!file.is_open()) {
            cerr << "Error opening file: " << fname << endl;
            return;
        }

        file >> n; // read number of chapters
        contents = new Content[n];
        for (int i = 0; i < n; ++i) {
            file.ignore(); // ignore newline character
            getline(file, contents[i].name);
            file >> contents[i].pno;
        }
        file.close();
    }

    ~TableOfContents() {
        delete[] contents;
    }

    void printContents() const {

```

```

        for (int i = 0; i < n; ++i) {
            cout << contents[i].name << " - Page " <<
contents[i].pno << endl;
        }
    }
};

// (b)
class Date {
private:
    int day, month, year;
public:
    Date(int d, int m, int y) : day(d), month(m), year(y) {}

    int operator-(const Date& other) const {
        // Calculate difference between two dates in days
        // Assume simple calculation for demonstration
        return (year - other.year) * 365 + (month -
other.month) * 30 + (day - other.day);
    }
};

class Book {
protected:
    string ISBN;
    string title;
public:
    Book(const string& isbn, const string& t) : ISBN(isbn),
title(t) {}

    virtual void printTableOfContents() const = 0;

    virtual void printDescription() const = 0;

    virtual void issueBook(const Date& issuanceDate) = 0;

```

```
virtual int calculateFine(const Date& returnDate) const =
0;
};

class ReferenceBook : public Book {
public:
    ReferenceBook(const string& isbn, const string& t) :
Book(isbn, t) {}

    void printTableOfContents() const override {
        cout << "Number of chapters: " << 2 << endl; // Assume
always 2 chapters for reference books
    }

    void printDescription() const override {
        cout << "Title is a list of words and their meanings"
<< endl;
    }

    void issueBook(const Date& issuanceDate) override {
        // Reference books cannot be issued
        cout << "Reference books cannot be issued." << endl;
    }

    int calculateFine(const Date& returnDate) const override {
        // Reference books have no fine
        return 0;
    }
};

class IssuableBook : public Book {
private:
    → Date issuanceDate;
    mutable int fine; // mutable to allow modification in const
method
public:
```

```

    IssuableBook(const string& isbn, const string& t, int d,
int m, int y) : Book(isbn, t), issuanceDate(d, m, y), fine(0)
{}

    void printTableOfContents() const override {
        // Implement print of all chapter names and page
numbers for issuable books
        cout << "Printing table of contents for " << title <<
endl;
    }

    void printDescription() const override {
        cout << "Title contains articles on various topics" <<
endl;
    }

    void issueBook(const Date& issuanceDate) override {
        this->issuanceDate = issuanceDate;
    }

    int calculateFine(const Date& returnDate) const override {
        int daysDiff = returnDate - issuanceDate;
        if (daysDiff > 14) {
            fine = 10 * (daysDiff - 14);
        }
        else {
            fine = 0;
        }
        return fine;
    }
};

// (c)
class Library {
private:
    Book** shelf1;

```

```

    ReferenceBook** shelf2;
    int count1, count2;
public:
    Library() : shelf1(nullptr), shelf2(nullptr), count1(0),
count2(0) {}

    void printBookDescriptions() const {
        cout << "Reference Book Descriptions:" << endl;
        for (int i = 0; i < count2; ++i) {
            shelf2[i]->printDescription();
        }
    }

    // Add methods to add books to shelves and manage counts
};

int main() {
    // Example usage
    TableOfContents toc("contents.txt");
    toc.printContents();

    ReferenceBook refBook("123456", "Dictionary");
    refBook.printDescription();

    IssuableBook textBook("789012", "Textbook", 1, 1, 2022);
    Date returnDate(31, 1, 2022);
    →textBook.issueBook(Date(1, 1, 2022));
    int fine = textBook.calculateFine(Date(16, 1, 2022));
    cout << "Fine for late return: Rs." << fine << endl;

    Library library;
    library.printBookDescriptions();

    return 0;
}

```

```
Microsoft Visual Studio Debug Console
Introduction - Page 5
Chapter 1 - Page 10
Chapter 2 - Page 20
Title is a list of words and their meanings
Fine for late return: Rs.10
Reference Book Descriptions:

D:\Filling With Classes\x64\Debug\Filling With Classes.exe
Press any key to close this window . . .
```

```
File Edit View
3
Introduction
5
Chapter 1
10
Chapter 2
20
```

- The first line 3 indicates the number of chapters.
- The next lines contain pairs of chapter names and page numbers:
 - "Introduction" starts on page 1.
 - "Chapter 1: Basics" starts on page 10.
 - "Chapter 2: Advanced Topics" starts on page 25.