



Улица Трета Македонска Бригада Скопје

Object Programing

Trial Final Exam

Exercise 1 (100p.): Library Book Management System

You are tasked with designing a simple C++ application for a library management system. The system will manage books, handle issued items and track usage based on daily limits.

Part 1: Base Class – Book (15%)

Create a class named Book with the following public data members:

- title (type: std::string)
- pages (type: int)
- Add a virtual method named display()

Part 2: Derived Class – IssuedBook (15%)

Create a class named IssuedBook that inherits from Book.

Add one additional public data member:

- borrowerName (type: std::string)

Override the display() method to print the book title, number of pages, name of the person who borrowed the book.

Part 3: Limit of Pages per Day (15%)

Define a constant:

- `const int MAX_PAGES_PER_DAY = 1000;`

Write logic to:

- Accumulate the total number of pages from issued books for a day.
- Display a warning if the total pages exceed the allowed maximum.

Part 4: Main Application Logic (40%)

In your main() function:

1. Create at least three instances of Book, such as:
 - *C++ Programming* (500 pages)
 - *History of Mathematics* (300 pages)
 - *Data Structures* (450 pages)
2. Create at least two instances of IssuedBook, such as:
 - *Clean Code* (350 pages, borrowed by Alice)
 - *Modern Physics* (400 pages, borrowed by Bob)
3. Store all books in a single list using pointers of type Book*, demonstrating Upcasting.
4. Use polymorphism to call display() and print information for each item.
5. Use Downcasting (dynamic_cast) to detect which items are of type IssuedBook and print a special message for borrowed books.
6. Allow the user to simulate issuing books for the day by selecting from the list. Sum up the number of pages from issued books and warn if the daily page limit is exceeded.

Part 5: Bonus (15%)

- Display only issued books using downcasting.
- Use std::vector<Book*> to manage book items dynamically.
- Allow multiple book selections using a loop and user input.