

MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY

Department of Computer Science and Engineering (CSE)

CSE 206: Object-Oriented Programming Language Sessional I

Online 2

Time: 1 Hr 30 min	Full marks: 35
Create a file yourID_online2_CSE206.cpp.	

	<p>Suppose that you are making a library management system. However, due to lack of time, you decided to make it quite simplified. Here's the class structure you came up with for C++ implementation:</p> <pre>classDiagram class Person { +Int ID +string name, email +double contact +display() = 0 +add_member() = 0 } class Student_member { +int level +string dept +display() = {display all info} +add_member() = {create a new obj of this class, take input} } class Faculty_member { +string faculty_name +string designation +display() = {display all info} +add_member() = {create a new obj of this class, take input} } class Research_Faculty_member { +string research_domain +int num_of_publications +display() = {display all info} +add_member() = {create a new obj of this class, take input} } class Books { +Int ID +string title, author_name +int price, quantity +display() = {display all info} +add_book() = {add a book} } class Books_issue { +string date_issued +display() = {display all info} +issue() = {issue a book by creating an obj of this class} } class Books_return { +int days_held +display() = {display all info} +return_b() = {return a book by creating an obj of this class} } Person < -- Student_member Person < -- Faculty_member Faculty_member < -- Research_Faculty_member Books < -- Books_issue Books < -- Books_return</pre> <p>Take a look at the inheritance scenario given above. Note that all inheritances are public inheritance unless stated otherwise. The first row is the class name. The second, third, and fourth rows denote the private, protected, and public members of the class respectively.</p> <p>Now, answer the following questions:</p>	
1	Implement the scenario depicted above in C++. Remember to write a parameterized constructor and destructor for each base and derived class. The constructor sets values for ALL the member variables. Also, implement methods (class functions) as described in the box.	10
2	Identify pure virtual functions, abstract class, and diamond problem in the above scenario. Address them accordingly when you implement the classes.	6
3	<p>Now, write the necessary code so that the appropriate display() function is called when the following code is added to your program.</p> <pre>void show_mem_info(Person *p) { p->display(); } int main() { Faculty_member f; f.add_member(); show_mem_info(&f); Student_member s; s.add_member(); show_mem_info(&s); }</pre>	4

3

Now, **overload** the following **operators** based on the details in the table given below. **Note that book is an object of Books and book_returned is an object of Books_return.**

8

Operator	Sample Operation	Description
++	book++	Increases the quantity of the book by 1.
+=	book+=5;	Increases the quantity of the book by 5.
-	book = 5 - book;	decreases the quantity of the book by 5.
N/A	int fine = book_returned;	if days_held > 30 then fine = (days_held-30) * 3 else fine = 0;

7

Now, create a menu in the main function the has the following features.

7

Menu Option	Feature Description
1 Add Member	Asks the user what type of member it is. Then takes input accordingly to create an object of that class. Then appends the info in a file named “member_log.txt”
2 Add Book	Adds a book by taking necessary input from the user. Creates an object and appends the info in a file named “book_log.txt”
3 Issue book	Calls the issue() function of Books_issue to create an object.
4 Return book	Calls the return_b() function of Books_return to create an object.
5 Exit	Closes the program