

Thapar Institute of Engineering & Technology, Patiala

Department of Computer Science and Engineering

MID Semester Test (MST)

B.E. (III Semester)	Time: 01.00 PM
Course Name (Code): Object Oriented Programming (UTA018)	Duration: 02 Hours
	M. Marks: 29+01 = 30
Instructor(s) (Lecture Groups): Neenu Garg (2C3, 2Q2), Seemu Sharma (2C2, 2Q1), Lovi Dhamija (2C5, 2Q3), Deep Mann (2W, 2I), Harpreet Singh (2C8, 2Q4), Subhendru Chattopadhyay (2C4, 2C7), Suresh Raikwar (2C1, 2C6), Shruti Aggarwal (2P at DeraBassi)	

Instructions: Attempt All questions. Marks are mentioned against each question. **One mark is reserved for attempting the questions in the order in which they are given. This mark will be deducted if the questions are not attempted in the given order. Plan the presentation of answers before writing.** Assume missing data, header file, namespaces, if any, suitably.

QN	Questions	Mark	CO/BL
Q1	In C++, both a "struct" and a "class" can support abstraction, encapsulation, inheritance, and polymorphism. If both can provide the major features of OOP, why does C++ provide both "struct and class"? Explain the key differences between them and illustrate your answer with a small example.	4	1/2
Q2	A. Predict and explain the output of the following program. <pre> 1. int* foo(int*& a, int& at) 2. {a = &at; return &at; } 3. int main() 4. { 5. int a = 10, b = 20, c = 30; 5. int* p = &a; 6. cout << "*p=" << *p << endl; 7. int* x = foo(p, b); 8. cout << "*p=" << *p << endl; 9. *x = 99; 10. cout << "b: " << b << ", c: " << c << endl; 11. return 0; 12. } </pre> B. "C does not provide classes, access specifiers, or abstract classes. Does that mean C cannot support encapsulation and abstraction?" Justify your answer by providing an example.	3+3	1,2/3
Q3	An AI tool is given the prompt: "Write a C++ class that provides the best possible data hiding for a bank account, with minimal latency overhead" to generated the following code snippet. <pre> class Account { double balance; public: double getBalance () { return balance; } void setBalance (double b) { balance = b; } }; </pre> The bank's requirements are: R1: The account balance must remain hidden from direct external access. R2: The balance may be modified only through setBalance (). The supplied amount is valid only if it is greater than zero. When a valid amount is supplied, it must be added to the existing balance, rather than replacing the existing balance. A. State whether the AI-generated code satisfies R1 and R2? Identify the requirement(s) that are not satisfied and explain why? B. Make the minimum modification to the body of setBalance () by adding statements only, so that the class satisfies R1 and R2. Give the modified code snippet.	2+2	1/3

Q4	Consider the following simple C++ class hierarchy. <table><tr><td><pre>#include <iostream> using namespace std; class Base { protected: int val; public: Base(int v) : val(v) {} void show() const { cout << "Base val: " << val << "\n"; } }; class Left: public Base { protected: int x; public: Left(int v, int xx) : Base(v), x(xx) {} void show_x() const { cout << "Left x: " << x << "\n"; } };</pre></td><td><pre>class Right: public Base { protected: int y; public: Right(int v, int yy) : Base(v), y(yy) {} void show_y() const { cout << "Right y: " << y << "\n"; } }; class Derived: public Left, public Right { <Insert the MISSING CODE here> }; int main() { Derived obj(10, 20, 30); return 0; }</pre></td></tr></table> A. Insert the missing code so that the program compiles successfully. B. Now, suppose statement obj.show(); is inserted in <i>main()</i>. Does the program still compile? If not, which line fails and why? Fix the issues in the code and rewrite the correct code.	<pre>#include <iostream> using namespace std; class Base { protected: int val; public: Base(int v) : val(v) {} void show() const { cout << "Base val: " << val << "\n"; } }; class Left: public Base { protected: int x; public: Left(int v, int xx) : Base(v), x(xx) {} void show_x() const { cout << "Left x: " << x << "\n"; } };</pre>	<pre>class Right: public Base { protected: int y; public: Right(int v, int yy) : Base(v), y(yy) {} void show_y() const { cout << "Right y: " << y << "\n"; } }; class Derived: public Left, public Right { <Insert the MISSING CODE here> }; int main() { Derived obj(10, 20, 30); return 0; }</pre>	2+3	2/3		
<pre>#include <iostream> using namespace std; class Base { protected: int val; public: Base(int v) : val(v) {} void show() const { cout << "Base val: " << val << "\n"; } }; class Left: public Base { protected: int x; public: Left(int v, int xx) : Base(v), x(xx) {} void show_x() const { cout << "Left x: " << x << "\n"; } };</pre>	<pre>class Right: public Base { protected: int y; public: Right(int v, int yy) : Base(v), y(yy) {} void show_y() const { cout << "Right y: " << y << "\n"; } }; class Derived: public Left, public Right { <Insert the MISSING CODE here> }; int main() { Derived obj(10, 20, 30); return 0; }</pre>						
Q5	A company wants to develop an Employee Salary Management System to calculate the monthly salary of its employees. The company maintains basic information such as employee ID, employee name, and basic salary. It also provides a house rent allowance (HRA) and medical allowance (MA) to every employee. Apply the concept of single inheritance to design and implement a C++ program for the above real-life scenario according to the following requirements: • Create a base class Employee containing the private data members: employeeID, employeeName, and basicSalary • Create a derived class Salary that publicly inherits from Employee. The derived class should contain HRA, MA, and grossSalary. Assume, HRA = 20% of basic salary, MA = 10% of basic salary and gross Salary = basicSalary + HRA + MA • The program should display the employee details along with the calculated gross salary.	05	2/2,3				
Q6	A. Identify the compilation error in the following code snippet and correct it by adding the statements only. Do not modify or remove any existing statement, member, access specifier, or class definition. <table><tr><td><pre>class Wallet; class Bank { int reserves = 100; friend class Wallet; public: void checkWallet (Wallet& w); };</pre></td><td><pre>class Wallet { int cashAmount = 200; }; void Bank::checkWallet(Wallet& w) { cout << w.cashAmount; }</pre></td></tr></table> B. Determine the output produced by the following code snippet and explain the reason for the output. <table><tr><td><pre>class A { static int x; int y; public: A(int v):y(v){ x += y;} static int getX(){ return ++x; } };</pre></td><td><pre>int A::x; int main() { cout << A::getX() << " "; A a1(3); cout << a1.getX() << " "; }</pre></td></tr></table>	<pre>class Wallet; class Bank { int reserves = 100; friend class Wallet; public: void checkWallet (Wallet& w); };</pre>	<pre>class Wallet { int cashAmount = 200; }; void Bank::checkWallet(Wallet& w) { cout << w.cashAmount; }</pre>	<pre>class A { static int x; int y; public: A(int v):y(v){ x += y;} static int getX(){ return ++x; } };</pre>	<pre>int A::x; int main() { cout << A::getX() << " "; A a1(3); cout << a1.getX() << " "; }</pre>	2+3	2/2,3
<pre>class Wallet; class Bank { int reserves = 100; friend class Wallet; public: void checkWallet (Wallet& w); };</pre>	<pre>class Wallet { int cashAmount = 200; }; void Bank::checkWallet(Wallet& w) { cout << w.cashAmount; }</pre>						
<pre>class A { static int x; int y; public: A(int v):y(v){ x += y;} static int getX(){ return ++x; } };</pre>	<pre>int A::x; int main() { cout << A::getX() << " "; A a1(3); cout << a1.getX() << " "; }</pre>						