



Daffodil International University
Faculty of Science & Information Technology
Department of Software Engineering
Midterm Examination, Summer 26

Course Code: SE216; Course Title: Object Oriented Programming

Sections & Teachers: UKD(A,B,C), RJM(D,E,H), MBH(F), MSN(G), AJE(I), MIH(J)

Time: 1 Hour 30 Mins

Marks: 25

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1. <i>or</i>	Convert the below JAVA Code to the appropriate Class Diagram. <pre> public class Movie { protected String title; void showMovie() { System.out.println("Movie: " + title); } } public class Horror extends Movie { void showMovie() { System.out.println("Horror Movie: " + title); } protected void showHorror() { System.out.println("Scare Level: Goosebumps "); } } public class Thriller extends Movie { String suspenseLevel; public void showThriller() { System.out.println("Suspense Level: " + suspenseLevel); } } public class SupernaturalHorror extends Horror { String ghostType; protected void showHorror() { System.out.println("Supernatural Horror Level: creepy"); } } public class PhysicalHorror extends SupernaturalHorror { private String injuryType; public void showInjury() { System.out.println("Physical Injury: " + injuryType); } } </pre>	[5]	CLO -2 Level -2
b)	A university is developing an employee management program. The system is divided into two packages named pack1 and pack2. Rewrite & design the classes using correct access modifiers. Package 1: pack1 holds class Employee 1. Class level: The class should be accessible from other packages. 2. Variables: • name: accessible only inside the same package • salary: accessible within the same package and in subclasses outside the package • id: should be fully hidden and only accessible inside the class • university: accessible from everywhere 3. Method: showDetails(): display all employee information and be accessible from other packages. Package 2: pack2 holds class Main Tasks: Inside the Main class • Create an object(non-parameterized) of Employee named spy • Try to access all variables directly: variables (name, salary, id, university) • Call showDetails() method	[5]	

	Observe and write: • which ones are accessible from the class Main • which ones give error • why they are not accessible		
c)	Compute the output of the following code: <pre> public class Argentina { int score = 3; int bonus; static double rating = 10.5; public Argentina() { System.out.println("3 world cups"); } public Argentina(int x) {} public void matchStart() { rating = 15.5; score = 6; System.out.println("Score: " + score); System.out.println("Rating: " + rating); } public void playGame() { int bonus = 50; System.out.println("Bonus Points: " + bonus); System.out.println("Score: " + score); rating = 20.0; } } </pre>	<pre> public void showStats() { System.out.println("Score " + score); System.out.println("Bonus " + bonus); System.out.println("Rating: " + rating); } public static void main(String[] args) { Argentina p1 = new Argentina(); Argentina p2 = new Argentina(4); p2.matchStart(); p1.playGame(); p1.matchStart(); p2.showStats(); } </pre>	[5]
2.	Observe the given code below to write a Player & StarPlayer class (Design class) to get the output as shown. Here, you have to deal with Upcasting, constructor overloading and method overloading, Inheritance. // Rewrite your design codes here: Driver Code: <pre> public class Main { public static void main(String[] args) { Player p1 = new Player("Messi", "Playmaker"); Player p2 = new StarPlayer(); StarPlayer p3 = new StarPlayer("Neymar"); p1.showInfo(); p2.showInfo(); p3.showInfo(); p1.updateStats(7); p2.updateStats(5); p3.updateStats("Neymar", "Winger"); p1.updateStats("Captain"); p1.Skill(); p3.specialSkill(); } } </pre> Expected Output: Messi is the star player Unknown Star Player Neymar is the star player Messi scored 7 goals Unknown Player scored 5 goals Neymar played as winger Messi played as captain A Magical dribbler A Clinical Finisher		[10] CLO -1 Level -2