

## Assignment-6 Classes and Objects

**Subject: Computer Science Workshop - 1 (CSE 2141)**

**Session: Sep 2025 to Jan 2026**

**Branch: Computer Science and Engineering (CSE)**

**Section: All**

**Course Outcome: CO1, CO2, CO3, CO4**

**Program Outcomes: PO1, PO2, PO3, and PO5**

**Learning Levels: Remembering (L1), Understanding (L2), Application (L3), Analysis (L4)**

<b>Q no.</b>	<b>Questions</b>	<b>Learning Levels</b>
Q1.	Write a Python program using a <b>Student</b> class that stores a student's name and roll number. The program should allow the user to enter how many students they want to add, take the details (name and roll) for each student through keyboard input, create a <b>Student</b> object for every entry, store all objects inside a list, and finally display the details of all students using a loop.	L1, L2
Q2.	Create a class called <b>Animal</b> with an <code>__init__</code> method. The constructor should take name and species as arguments and assign them to instance variables of the object. Add a <code>__str__</code> method to the class which should return a string in the format "Lucy(Cat)". When an <b>Animal</b> object is printed, it should display the name and species in a readable format.	L1, L2
Q3.	Write a Python program using two classes, <b>Person</b> and <b>College</b> , to demonstrate object composition. The <b>College</b> class should create two <b>Person</b> objects inside its constructor, one with gender "female" and the other with "male". Create a <b>College</b> object and print the gender of both students.	L2, L3
Q4.	Write a Python program that defines a <b>Car</b> class with two attributes, <b>make</b> and <b>model</b> , and initialize them using a parameterized constructor ( <code>__init__(self, v1, v2)</code> ). Include getter and setter methods to access and modify these values. In the main part of the program, create two <b>Car</b> objects, one initialized with valid values and another initialized with <b>None</b> , display their details using the getter methods, then update the second object using the setter methods, and print the updated result.	L2, L3
Q5.	Write a Python program to create an <b>Image</b> class with attributes <b>imageWidth</b> , <b>imageHeight</b> , and <b>colorCode</b> . The class should include a constructor ( <code>__init__</code> ) to initialize these values, setter methods to update them, getter methods to retrieve them, and an overridden <code>__str__</code> method to display the image details in a readable format. Create two objects of the <b>Image</b> class, one using a default constructor and another using a parameterized constructor, and print the details of both objects.	L2, L4

Q6.	<p>Write a Python program to demonstrate the role of the <code>__new__</code> method in object creation, especially when working with immutable objects. Create a class that overrides <code>__new__</code> to print a message, create the instance manually using <code>object.__new__(cls)</code>, and then use <code>__init__</code> to print another message during initialization. Also show how <code>__new__</code> can be used to modify the value of an immutable object (such as an integer) at creation time. Create objects and observe how <code>__new__</code> executes before <code>__init__</code> and how it controls the creation of immutable types.</p>	L3, L4
Q7.	<p>Design a Python program to simulate a ticket booking system for a cinema hall. Each ticket should have attributes such as movie name, show time, seat number, and ticket price. Use a constructor to initialize these attributes when creating a ticket. Implement methods to display ticket details and calculate the total price for multiple tickets. Simulate booking multiple tickets, display each ticket's details, and show the total amount to be paid.</p>	L2, L3
Q8.	<p>Design a Product class with name, price, and quantity attributes. Implement a class variable <code>total_products_sold</code> that keeps track of the total number of products sold across all instances. Create an instance method <code>sell_product(amount)</code> that decrements the quantity and increments <code>total_products_sold</code>. Also, implement a class method <code>get_total_products_sold()</code> that returns the value of <code>total_products_sold</code>.</p>	L2, L4
Q9.	<p>Create a Python program for a university student record system using encapsulation. Define a <b>Student</b> class with attributes <code>name</code>, <code>roll_no</code>, and a private variable <code>__marks</code>. Provide a setter method that assigns marks only if they are between 0 and 100, and a getter method to access them. Based on the marks, automatically calculate the grade (A, B, C, D, E, or F). Include a method <code>show_details()</code> to display the student's name, roll number, marks, and grade. Create multiple student objects, set their marks using the setter method, and display their details.</p>	L3, L4
Q10.	<p>Design a Python class to represent a book in a library system, storing details such as title, author, ISBN number, availability status, and the number of copies. Include methods to issue a book (decrease copies), return a book (increase copies), and check if a book is available. Finally, create multiple book objects and simulate issuing and returning books to demonstrate how the system works.</p>	L3, L4
	<b>-END-</b>	