Course Code: CSE 2102
Course Title: Object Oriented Design and Design Patterns

Topic: Taking user input & method creation.

1. Write a Java program to take your name, id, semester, section as input. Use show method to display this information.
2. Write a Java program to perform addition, subtraction, multiplication, division between two numbers by creating user defined method.

Topic: Problem Solving (conditional statements & loop statement)

1. Write a Java program to swap two numbers.
2. Write a Java program to find maximum of three numbers.
3. Write a Java program to find whether a given number is even or odd.
4. Write a Java program to identify whether a character is uppercase, lowercase, digit, or special character.
5. Write a Java program to generate multiplication table for a given number.
6. Write a Java program to find all number that is divisible by 7 between a given intervals.
7. Write a Java program to count the number of digits of an integer value.
8. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.

Course Code: CSE 2102

Course Title: Object Oriented Design and Design Patterns

Topic: Problem solving (Array)

1. Write Java program to find the sum of all odd numbers in an array.
2. Write a java program that takes two arrays as input, calculate the index wise sum of these arrays, and store the result in a third array.
3. Write a Java program to search an element in an array.
4. Write a Java program to reverse the elements in an array without using a second array.
5. Write a Java program to find second highest element of an array.
6. Write a Java program that calculates the average of an array, excluding the highest and lowest values in the array.
7. Write a Java program to find the sum of all the prime numbers in a 2d array.

Topic: OOP concepts- Class, Object & Constructor

1. Create a Grader class with an instance variable marks, and a letterGrade() method to return the letter grade based on the marks. Create an object of Grader class and ensure that the marks is not negative and is not greater than 100.
2. Write a program to create an Employee class that includes the attributes for employee_id, employee_name, basic_salary, bonus, overtime_pay, insurance, and taxes. Provide appropriate methods for calculating net salary and displaying information.

   Net salary = (Basic + Bonus + Overtime) – (Insurance + Tax)
3. Write a Java program for a Candidate class with attributes RNo, name, score, and remarks. Include methods assign_remarks() to set remarks based on score (≥50 = "selected", else "not selected") and display() to print details.

4. Write a Java program to create a Car class with instance variables brand, color, and year. Implement a parameterized constructor to initialize these variables. Use a function to display these information.
5. Create a Student class with attributes for name, semester, and courses. Include methods for adding and removing courses.

Topic: OOP concepts- Constructor & Constructor Overloding

1. Create an Account class with two private data members: Account_No and Balance. The class should also include a static data member C to track the total number of accounts created. In the main method, create three account objects, calculate and display the average balance of these accounts.
2. Create a Java program with a class named BankAccount. The class should contain fields for the account holder's name, address, account number, and balance. Include a method to display account details, a method to deposit amount, and a method to withdraw amount.
3. Create a class named Circle with two private instance variables: radius and color. Initialize these variables with default values of 1.0 and "red", respectively, using a default constructor. Include a second constructor that accepts a parameter for radius and uses the default value for color. Provide a public method getArea(), to return the area of the circle.
4. Create a class named Book with two private instance variables: title and author. Initialize these variables with default values using a default constructor. Include a second constructor that accepts parameters for title and author. Provide a public method show(), to print the book details.

Course Code: CSE 2102
Course Title: Object Oriented Design and Design Patterns

Topic: OOP concepts- Encapsulation & Inheritance

1. Create a Rectangle class with private instance variables length and width. Include appropriate methods to access and modify these variables. Also, include methods to calculate area and perimeter of a rectangle.
2. Create an Account class with private instance variables account_number, name, and balance. Provide public getter and setter methods to access and modify these variables.
3. Create a base class Building that stores the number of floors of a building and, number of rooms. Create a derived class House that inherits Building and also stores the number of bedrooms and kitchens. Demonstrate the working of the classes.
4. Create a base class called "vehicle" that stores number of wheels and speed. Create the following derived classes – "car" that inherits "vehicle" and also stores number of passengers. "truck" that inherits "vehicle" and also stores the load limit. Write a main function to create objects of these two derived classes and display all the information about "car" and "truck". Also compare the speed of these two vehicles - car and truck and display which one is faster.

Course Code: CSE 2102
Course Title: Object Oriented Design and Design Patterns

Topic: OOP concepts- Polymorphism

1. Create a Java class Calculator that provides different ways to perform addition. Include three methods: The first method takes two integer numbers, the second method takes three integer numbers, and the third method takes two double numbers. In the main method, create an object of Calculator class.
2. Create a Java class Shape that provides different ways to calculate the area. Include three methods: the first method takes one parameter (side length) to calculate the area of a square, the second method takes two parameters (length and width) to calculate the area of a rectangle, and the third method takes one decimal parameter (radius) to calculate the area of a circle. In the main method, create an object of Shape class.
3. Write a Java program to define a class Employee with instance variables name and id, along with a method calculateSalary(). Create two subclasses, Worker and Supervisor, each having additional instance variables baseSalary and bonus. In both subclasses, override the calculateSalary() method to compute and return the salary.
4. Write a Java program to define a class Vehicle with a method speedUp(). Create two subclasses: Car and Bicycle, each having an instance variable currentSpeed. In both subclasses, override the speedUp() method to increase the vehicle's speed differently.

Course Code: CSE 2102
Course Title: Object Oriented Design and Design Patterns

Topic: OOP concepts - Abstraction

1. Write a Java program to create an abstract class Shape with abstract methods getarea() and getperimeter(). Create subclasses Rectangle and Circle that extend the Shape class and implement all the abstract methods.
2. Write a Java program with an abstract class BankAccount containing variables accountNumber, accountHolderName, and balance, along with abstract methods deposit() and withdraw(). Create a subclass SavingsAccount that implements these abstract methods. Demonstrate the functionality by performing deposits and withdrawals, then displaying the updated balance.
3. Create an interface named **Department** with variables(as constants) ‘deptName’ and ‘head’ including abstract methods such as:
    - printDeptName()
    - printDeptHead()

   Create a class named **Office** with attributes ‘officeName’, ‘officeLocation’, and ‘numberOfEmployees’ and include methods such as:
    - setOfficeInfo()
    - getOfficeName()
    - getOfficeLocation()
    - getNumberOfEmployees()
    - displayOfficeDetails()

   Create a class named **Employee** extending **Office** and implementing the **Department** interface (by overriding the abstract methods). This class should include attributes like ‘employeeId’, ’employeeName’, ‘position’ and include methods such as:
    - setEmployeeInfo()
    - getEmployeeId()
    - getEmployeeName()
    - getPosition()

- displayEmployeeDetails()

Now, in the main() method create objects of both Employee and Office class and invoke the functions.