

Government Polytechnic Mandkola Palwal

Lesson Plan

Name of Faculty : Rahul Kaushik
Discipline :COMPUTER ENGG.
Semester :4TH SEM
Subject :OOPS using Java
Lesson Plan Duration : 45 lecturers (from 06 March 2023 to 23 June 2023)

Week	Theory		Practical	
	Lecture Day	Topic (Including Assignment / Test)	Practical Day	Topic
I	1	Fundamentals of object oriented programming – procedure oriented programming Vs. object oriented programming (OOP)	1	Consider we have a Class of Cars under which Santro Xing, Alto and Wagon R represents individual Objects. In this context each Car Object will have its own, Model, Year of Manufacture, Colour, Top Speed, etc. which form Properties of the Car class and the associated actions i.e., object functions like Create(), Sold(), display() form the Methods implementation of keywords like try, catch, finally, throw & throws. importance of exception handling in practical implementation of live projects.of Car Class. Use this class to create another class Company that tracks the models it create.
	2	Object oriented programming concepts – Classes, object, object reference,	2	-d0-
	3	Abstraction, Encapsulation,		
2	1	Inheritance, Polymorphism	1	software company Software Engineers, Sr. Software Engineers, Module Lead, Technical Lead, Project Lead, Project Manager, Program Manager, Directors all are the employees of the company but their work, perks, roles, responsibilities differs. Create the Employee base class would provide the common behaviors of all types of employee and also some behaviors

				properties that all employee must have for that company. Also include search method to search an employee by name.
	2	Introduction of eclipse (IDE) for developing programs in Java	2	-do-
	3	Review of constructs of C used in JAVA : variables, types and type declarations		
3	1	Data Types, Increment and Decrement operators, Relational and Logical operators;	1	Suppose the Airport personals want to maintain records for the arrival and departure of the planes. Create a class Airport that has data like name, id, and address. Create two more classes for Arrival and Departure implementing Airport that will have track of planes (their name, id, arrival time or departure time and a counter to count the number of arrivals) also include the necessary methods to access the information.
	2	if then else clause;	2	-do-
	3	Conditional Expressions,		
4	1	Input using scanner class and Output statement,	1	Create a whole menu driven hospital management system using concept of OOP like classes, inheritance. Include information about the following: <ul style="list-style-type: none"> a. Patient -name, registration id, age, disease, etc. b. Staff – id, name, designation, salary, etc.
	2	Loops, Switch Case,	2	-do-
	3	Arrays, Methods.		
5	1	Creation, accessing class members	1	Create a class called Musicians to contain three methods string (), wind () and perc ()

				<p>). Each of these methods should initialize a string array to contain the following instruments:</p> <ul style="list-style-type: none"> - veena, guitar, sitar, sarod and mandolin under string () - flute, clarinet saxophone, nadhaswaram and piccolo under wind () - tabla, mridangam, bangos, drums and tambour under perc ()
	2	Private Vs Public Vs Protected Vs Default	2	-do-
	3	Constructors Object & Object Reference		
6	1	Revision	1	Revision
	2	Revision	2	Revision
	3	Ist sessional		
7	1	Definition of inheritance,	1	Write three derived classes inheriting functionality of base class person (should have a member function that ask to enter name and age) and with added unique features of student, and employee, and functionality to assign, change and delete records of student and employee.
	2	Protected data,	2	-do-
	3	Private data, Public data,		
8	1	Constructor chaining,	1	Using the concept of multiple inheritance create classes: Shape, Circle, Square, Cube, Sphere, Cylinder. Your classes may only have the class variable specified in the table below and the methods Area and/or Volume to output their area and/or volume.

	2	Order of Invocation	2	-do-
	3	Types of Inheritance		
9	1	Single inheritance	1	<p>Write a program to create class Person.</p> <p>a. Make two classes, Student and Instructor, inherit from Person. A person has a name and year of birth.</p> <p>b. A student has a major, student id.</p> <p>c. An instructor has salary, subject.</p> <p>Write the class definitions, the constructors, set methods, get methods and for all classes.</p>
	2	Multilevel inheritance,	2	-do-
	3	Hierarchical inheritance,		
10	1	Hybrid inheritance	1	<p>Old MacDonald had a farm and several types of animals. Every animal shared certain characteristics: they had a type (such as cow, chick or pig) and each made a sound (moo, cluck or oink). An Interface defines those things required to be an animal on the farm. Define new classes for the Old MacDonald that implement the Animal and Farm class. Create array of object of animal to define the different types of animal in the farm. Also create appropriate methods to get and set the properties.</p>
	2	Method & Constructor	2	-do-
	3	Overloading, Method		

		overriding		
11	1	up-casting and down-casting.	1	Revision
	2	Key points of Abstract class & interface,	2	Revision
	3	Difference between an Abstract class & Interface,		
12	1	Implementation of multiple inheritance through interface.	1	Write a program with Student as abstract class and create derive classes Engineering, Medicine and Science from base class Student. Create the objects of the derived classes and process them and access them using array of pointer of type base class Student.
	2	Revision	2	-do-
	3	Sessional-2		
13	1	Definition of exception handling,	1	Revision
	2	Implementation of keywords like try catch	2	Revision
	3	Finally,		
14	1	Throw & throws..	1	Revision
	2	Importance of exception handling in practical implementation of live projects	2	Revision
	3	Revision		
15	1	Revision	1	Revision
	2	Revision	2	Revision
	3	Sessional 3		