CVM UNIVERSITY
Aegis: Charutar Vidya Mandal (Estd.1945)

FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2020-21

Programme: **Bachelor of Engineering (Information Technology)**

Semester: **IV**

Course Code: **102044502**

Course Title: **Programming with Java**

Course Group: **Professional Core Course**

Course Objectives: To understand the basic concepts and fundamentals of platform independent object-oriented language and gain knowledge about basic Java language syntax. To demonstrate skills in writing programs using exception handling techniques, multithreading, and File Management system. To Understands Collection framework and generic programming. To understand streams, network programming and efficient user interface design techniques**.**

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
3	0	2	4	40 / 14	60 / 21	20/7	30/10	150 / 52

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	**Introduction:** Feature of Java, Java Virtual Machine, Byte Code, JDK, JRE, Comments, Java coding convention **Array and String:** Single and Multidimensional Array, Command line argument, String, String Buffer, String Pool, for-each loop, var-length arguments. Wrapper Class, Arrays class **Class Object and Method:** Class, Object, Constructor, Method overloading, Constructor overloading, passing, and returning Object as function arguments, passing and returning array of object in function**,** new keyword, this keyword, static variable, static method, garbage collection, finalize (), access modifier (default, public, private, protected), Singleton class **Diagram:** Class Diagram, Object Diagram, Has-A Relation	**11**

2	**Inheritance: Inheritance:** Use of inheritance, Object Class, types of inheritance, Coupling-and-typecasting, Abstract class, Abstract method, Dynamic method dispatch, final keyword (variable, method, class), super keyword, Interface, interface naming conflicts, method naming conflict, variable naming conflicts, marker interface, interface vs abstract class vs concrete class **Diagram:** Generalization, Specialization, Relationship between classes (Is-A Relation)	**04**
3	**Inner Class:** Class inside Class, method local inner class, anonymous inner-class, interface inside class, interface inside interface, class inside interface, static nested class, adapter classes **Exception Handing:** Exception and Error, use of try, catch, throw, throws and finally, Built in Exception, Custom exception, Throwable Class Runtime Stack Mechanism, nested try **Package**: Use of Package, CLASSPATH, Import statements, Static import, Access control	**06**
4	**Threading:** Introduction, ways to define, instantiate and start a new Thread, Thread class constructors, Thread priority, yield(), join(), sleep(), Synchronization, Inter Thread communication, Deadlock, introduction of executor framework **Generics:** Generic clause, Bounded Type, Generic Methods, and wild char character(?), Communication with non-Generic code, comparable and comparator	**06**
5	**Collection API:** Collection Framework, ArrayList class, LinkedList class, ListIterator interface, HashSet class, LinkedHashSet class, TreeSet class, PriorityQueue class, ArrayDeque class, Map interface, HashMap class, LinkedHashMap class, TreeMap class, Hashtable class, Properties class, StreamTokenizer class	**03**
6	**File Handling Using Java:** Introduction to Stream, Byte Stream, Character stream, Readers and Writers, File Class, FileOutputStream, FileInputStream, FileWriter, FileReader, ,DataInputStream, DataOutputStream, InputStreamReader, Console, Scanner, PrintStream , PrintWriter class, BuffredReader , object Serialization	**03**
7	**Network Programming Using Java:** Introduction to Socket Programming, URL class, InetAddress class, DatagramSocket and DatagramPacket, ServerSocket, Socket class **GUI Programming:** Introduction of java, swing, Components, Event-Delegation-Model, Listeners, Layouts, Individual Components Label, Button, Check Box, Radio Button, Choice, List, Menu, Text Field, Text Area	**07**
	Total	**40**

List of Practicals / Tutorials:

1	**Basic Program** 1. Study of class path and java runtime environment 2. Write a program to ● Implement command line calculator ● Write To prints Fibonacci series.
2	**Array:**

1. Define a class Array with following member

Field:
int data[];
Function:
Array( ) //create array data of size 10
Array(int size) // create array of size size
Array(int data[]) // initialize array with parameter array
void Reverse _an _array () //reverse element of an array
int Maximum _of _array () // find maximum element of array
int Average_of _array() //find average of element of array
void Sorting () //sort element of array
void display() //display element of array
int search(int no) //search element and return index else return -1
int size(); //return size of an array

Use all the function in main method. Create different objects with different constructors.

2. Define a class Matrix with following

Field:
int row, column;
float mat[][]

Function:
Matrix(int a[][])
Matrix()
Matrix(int rwo, int col)
void readMatrix() //read element of array
float [][] transpose( ) //find transpose of first matrix
float [][] matrixMultiplication(Matrix second) //multiply two matrices and return result
void displayMatrix(float [][]a) //display content of argument array
void displayMatrix() //display content
float maximum_of_array() // return maximum element of first array
float average_of_array( ) // return average of first array

create three object of Matrix class with different constructors in main and test all the functions in main

3. Write a program to demonstrate usage of different methods of Wrapper class
4. Write a program to demonstrate usage of String and StringBuffer clas
5. Define a class Cipher with following data

Field:
String plainText;

	int key Functions: Cipher(String plaintext,int key) String Encryption( ) String Decryption( ) Read string and key from command prompt and replace every character of string with character which is key place down from current character. Example plainText = "GCET" Key = 3 Encryption function written following String " JFHW" Decryption function will convert encrypted string to original form "GCET"
3	**Basic Program using Class** 1. Create a class BankAccount that has Depositor name , Acc_no, Acc_type, Balance as Data Members and void createAcc() . void Deposit(), void withdraw() and void BalanceInquiry as Member Function. When a new Account is created assign next serial no as account number. Account number starts from 1 2. Create a class time that has hour, minute and second as data members. Create a parameterized constructor to initialize Time Objects. Create a member Function Time Sum (Time, Time) to sum two time objects. 3. Define a class with the Name, Basic salary and dearness allowance as data members.Calculate and print the Name, Basic salary(yearly), dearness allowance and tax deduced at source(TDS) and net salary, where TDS is charged on gross salary which is basic salary + dearness allowance and TDS rate is as per following table. DA is 74% of Basic Salary for all. Use appropriate member function.
4	**Inheritance and interface** 1. class Cricket having data members name, age and member methods display() and setdata(). class Match inherits Cricket and has data members no_of_odi, no_of_test. Create an array of 5 objects of class Match. Provide all the required data through command line and display the information. 2. Define a class Cripher with following data Field: String plainText; int key Functions: Cipher(String plaintext,int key) abstract String Encryption( )

Gross Salary	TDS
Rs. 100000 and below	NIL
Above Rs. 100000	10% on excess over 100000

abstract String Decryption( )

Derived two classes Substitution_Cipher and Caesar_Cipher override Encyption() and Decyption() Method. in substitute cipher every character of string is replace with another character. For example. In this method you will replace the letters using the following scheme.
Plain Text: a b c d e f g h i j k l m n o p q r s t u v w x y z
Cipher Text: q a z w s x e d c r f v t g b y h n u j m i k o l p

So if string consist of letter "gcet" then encrypted string will be "ezsj" and decrypt it to get original string
In ceaser cipher encrypt the string same as program 5 of LAB 5.

3. Declare an interface called Property containing a method computePrice to compute and return the price. The interface is to be implemented by following two classes i) Bungalow and ii) Flat.
Both the classes have following data members
   - name
   - constructionArea

The class Bungalow has an additional data member called landArea. Define computePrice for both classes for computing total price. Use following rules for computing total price by summing up sub-costs:
Construction cost(for both classes):Rs.500/- per sq.feet
Additional cost (for Flat) : Rs. 200000/-
(for Bungalow): Rs. 200/- per sq.
feet for landArea
Land cost (only for Bungalow): Rs. 400/- per sq. feet
Define method main to show usage of method computePrice.

4. Define following classes and interfaces.

```
public interface GeometricShape {
  public void describe();
}
public interface TwoDShape extends GeometricShape {
  public double area();
}
public interface ThreeDShape extends GeometricShape {
  public double volume();
}
public class Cone implements ThreeDShape  {
  private double radius;
  private double height;
  public Cone (double radius, double height)
  public double volume()
  public void describe()
```

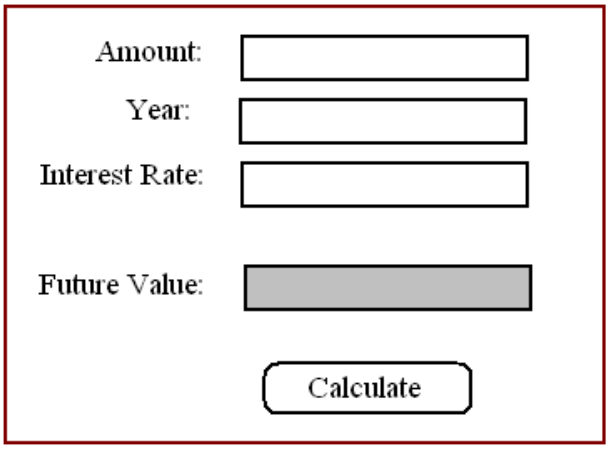
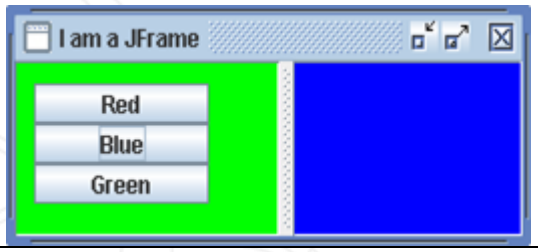
	}
	public class Rectangle implements TwoDShape {
	private double width, height;
	public Rectangle (double width, double height)
	public double area()
	public double perimeter()
	public void describe()
	
	}
	public class Sphere implements ThreeDShape {
	private double radius;
	public Sphere (double radius)
	public double volume()
	public void describe()
	}
	Define test class to call various methods of Geometric Shape

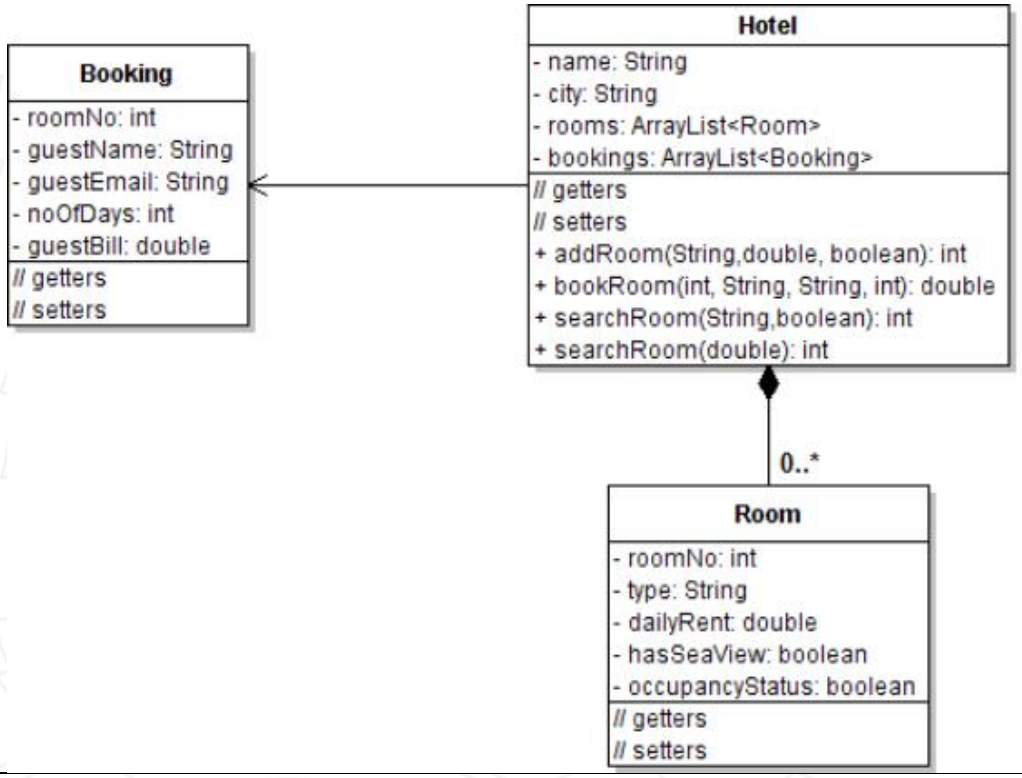
5	**Inner Class:**
	Define two nested classes: Processor and RAM inside the outer class: CPU with following data members
	class CPU {
	double price;
	class Processor{ // nested class
	double cores;
	double catch()
	String manufacturer;
	double getCache()
	void displayProcesorDetail()
	}
	protected class RAM{ // nested protected class
	// members of protected nested class
	double memory;
	String manufacturer;
	Double clockSpeed;
	double getClockSpeed()
	void displayRAMDetail()
	}
	}
	1. Write appropriate Constructor and create instance of Outer and inner class and call the methods in main function
	2. Write a program to demonstrate usage of static inner class, local inner class and anonymous inner class

6	**Generics** 1. Declare a class InvoiceDetail which accepts a type parameter which is of type Number with following data members class InvoiceDetail <N extends Number> { private String invoiceName;

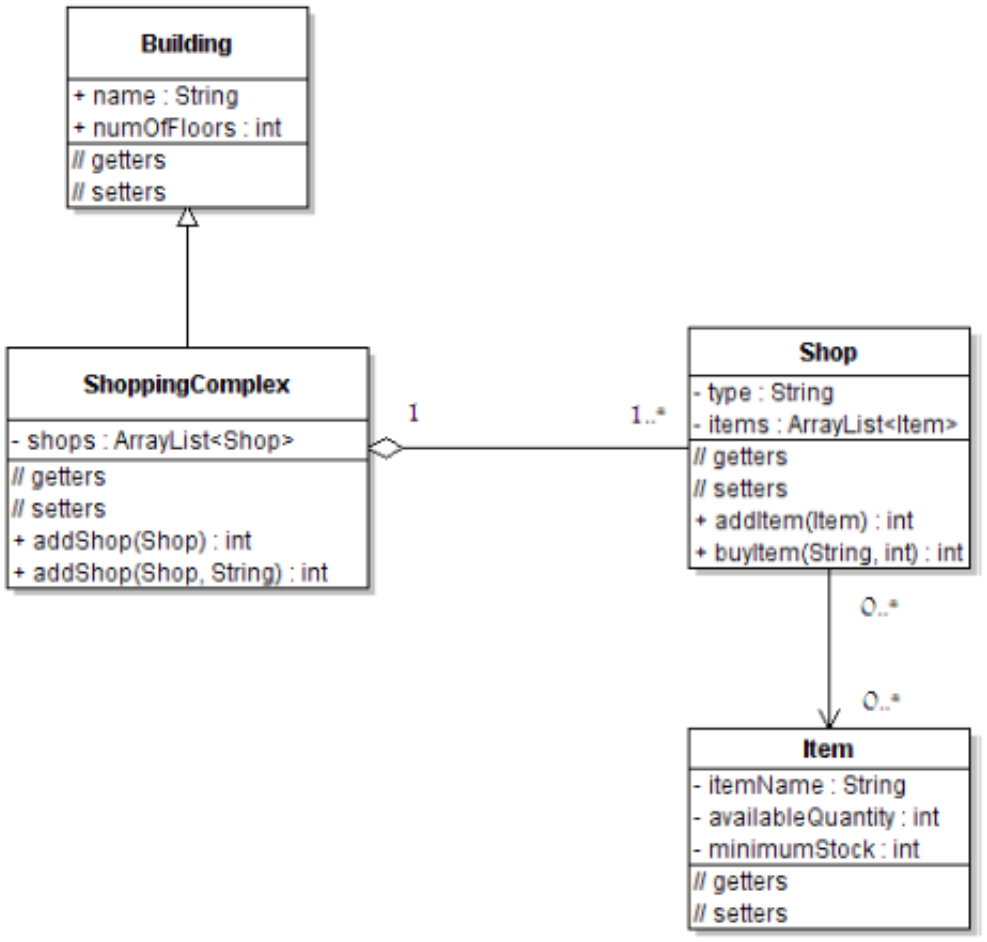
	private N amount;
	private N Discount
	// write getters, setters and constructors
	}
	Call the methods in Main class
	
	2. Implement Generic Stack
	3. Write a program to sort the object of Book class using comparable and comparator interface. (Book class consist of book id, title, author and publisher as data members)

7	**Exception Handing** 1. Write a program for creating a **Bank** class, which is used to manage the bank account of customers. Class has two methods, Deposit () and withdraw (). Deposit method display old balance and new balance after depositing the specified amount. Withdrew method display old balance and new balance after withdrawing. If balance is not enough to withdraw the money, it throws **ArithmeticException** and if balance is less than 500rs after withdrawing then it throw custom exception, **NotEnoughMoneyException**. 2. Write a complete program for calculation average of n +ve integer numbers of Array A. a. Read the array form keyboard b. Raise and handle Exception if i. Element value is -ve or non-integer. ii. If n is zero.
8	**Threading** 1. Write a program to find prime number in given range using both method of multithreading. Also run the same program using executor framework 2. Assume one class Queue that defines queue of fix size says 15. ● Assume one class producer which implements Runnable, having priority NORM_PRIORITY +1 ● One more class consumer implements Runnable, having priority NORM_PRIORITY-1 ● Class TestThread is having main method with maximum priority, which creates 1 thread for producer and 2 threads for consumer. ● Producer produces number of elements and put on the queue. when queue becomes full it notifies other threads. Consumer consumes number of elements and notifies other thread when queue become empty.
9	**Collection API:** 1. Write a program to demostrate user of ArrayList, LinkedList ,LinkedHashMap, TreeMap and HashSet Class. And also implement CRUD operation without database connection using Collection API. 2. Write a program to Sort Array,ArrayList,String,List,Map and Set

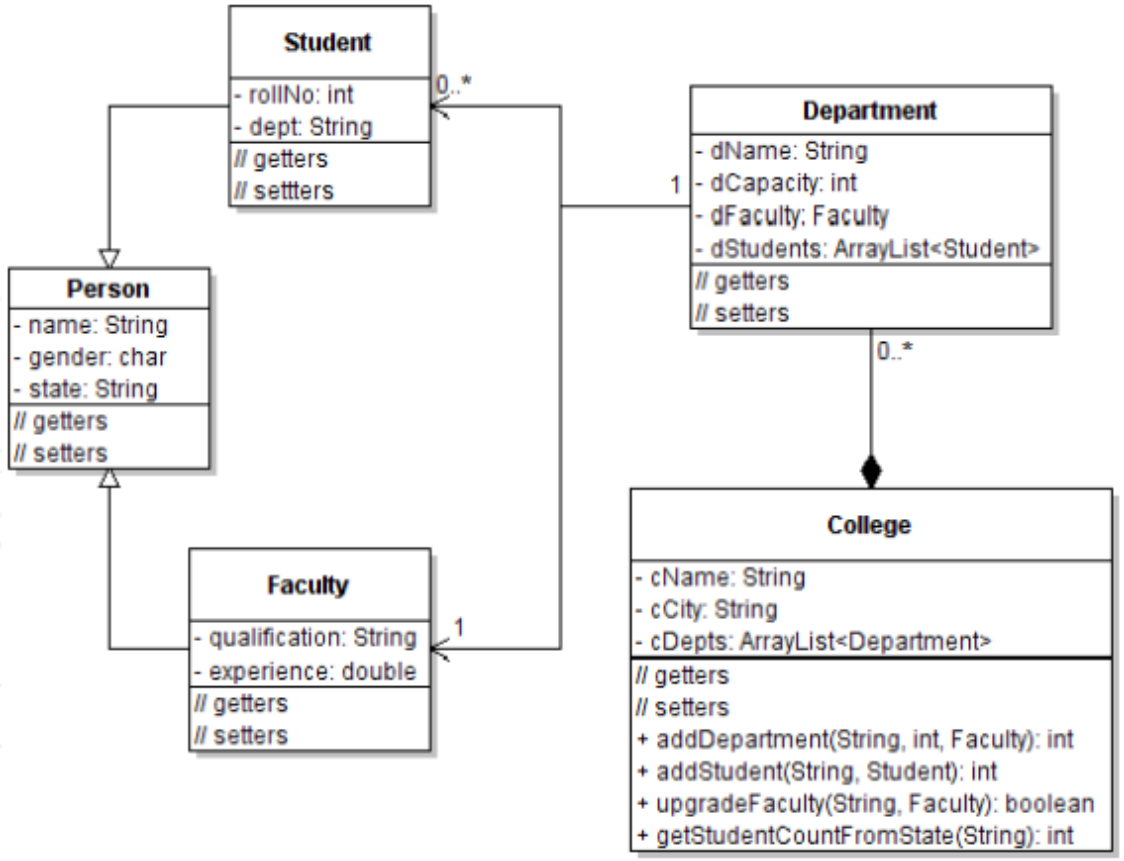
10	**File Handling Using Java:** 1. Write a programme to count occurrence of a given words in a file. 2. Write a program to print it seltf. 3. Write a program to display list of all the files of given directory
11	**Networking** 1. Implement Echo client/server program using TCP 2. Write a program using UDP which give name of the audio file to server and server reply with content of audio file
12	**GUI** 1. Write a programme to implement an investement value calculator using the data inputed by user. textFields to be included are amount, year, interest rate and future value. The field "future value" (shown in gray) must not be altered by user.  2. Write a program which fill the rectangle with the selected color when button pressed. 
13	**Case Study 1:** **Seven Seas**, a reputed 3-star hotel of Gandhinagar, wants to automate its room management and booking services for providing better service to their customers. Define the classes as shown in the below class diagram, specifying all the getters and setters for each class. Booking object can only be created by specifying the room number, guest name, guest email id and number of days guest would be staying at the hotel. Room object can be created only by-passing room type ("S" - Standard, "D" - Deluxe and "L" - Luxury), daily rent and specifying whether the room has sea view or not. Room number will be assigned by the system as and when the room gets added (starting from room number 101). For every

	new room added, the default occupancy status will be set as false. Hotel object can be created without passing any arguments. Hotel class should be defined as public class and it has the following behaviors: 1) **addRoom:** This behavior accepts 3 arguments, i.e. room type, daily room rent and sea view flag. It assigns a room number to the added room and returns the room number. 2) **bookRoom:** This behavior accepts an unoccupied room number, guest's name, email id and the number of days the guest wants to stay at the hotel room (in given order). The behavior sets the bill attribute and returns the same. If the specified room is occupied, the behavior does not set the bill attribute and returns -1. 3) **searchRoom:** This behavior allows to search with two different criteria: a) It accepts the type of room desired and flag that specifies if sea view is also required. b) It accepts the upper limit the guest would like to pay as daily rent for a room For both (a) and (b) criteria, the behavior is supposed to return the number of unoccupied rooms matching the specified criteria. NOTE: Data and Business validations need not be handled. 
14	**Case Study-2 :** **Shopping Complex** A Shopping Complex is a building which has many shops. Each shop has many items to be sold. A customer can buy items from any shop in the shopping complex if stock is available. There are four classes defined: **Building, ShoppingComplex, Shop and Item** where ShoppingComplex is the public class. A Building object can only be

	created by passing its name and number of floors that it has. A ShoppingComplex object can be created by passing a Shop object to it. A Shop object can be created only by passing its type. An Item object can be created only by passing its name, available quantity and minimum stock. The business behaviors should be implemented as mentioned below: **addShop (Shop newShop) :** This behavior takes a Shop object and adds it to the list of shops in the ShoppingComplex if the list does not already have a shop of the same type. It returns the total number of shops in shopping complex. **addShop (Shop newShop, String type) :** This method adds a Shop object to the array list only if the type of the Shop matches with the type passed as input argument to this method and returns the total number of shops in shopping complex. If the type of Shop does not match with the type passed as input then this method should return -1. **addItem (Item newItem) :** This method adds Item object to the shop if that item name is not already present in that shop and returns the total number of items in the shop. If that item is already present in the shop it returns -2. **buyItem(String itemName, int requiredQuantity) :** This behavior takes an item name and required quantity. An item is sold if the remaining quantity of that item after that sale is >= the minimum stock for that item. In that case it returns the updated quantity of that item. If the given item name is not present in the shop then this behavior returns -1. In case the item is present but not available in required quantity then it returns -3. Note : Data and business validations are not required. Class Diagram:



15 **Case Study 3:**
NalSarovar college is a popular college in Gandhinagar city of Gujarat. The college has decided to automate its Department, Student and Faculty management activities so as to improve their operation efficiency. The proposed system needs to be developed as per the Class diagram shown below:



Person object can only be created by specifying person's name, gender(M / F) and the state he/she belongs to. Unless a student is added to a department, he/she will neither have roll number nor a department. Hence Student object can be created without passing these arguments. Every faculty has some qualification and certain experience. Hence, faculty object can be created only by passing these attribute values.

Department object can only be created by providing its name, student capacity and corresponding faculty in-charge. College object can only be created by specifying its name and city. College class to be defined as a public class.

Reference Books:

1	OCA Java SE 8 Programmer I Study Guide (Exam 1Z0-808) (Oracle Press)
2	OCP: Oracle Certified Professional Java SE 8 Programmer II Study Guide: Exam 1Z0-809
3	SCJP Sun Certified Programmer for Java 6 Study Guide: Exam 310-065
4	The class of JAVA by Pravin M. Jain
5	Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
6	The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, -TMH

Supplementary learning Material:	
1	Lecture Note, SWAYAM – NPTEL Course "Programming with Java"
2	Open-source Tools (Java 8, Visual Studio Code, eclipse)

Pedagogy:

- Direct classroom teaching
- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation
- Industrial/ Field visits
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding; **A**: Applying; **N**: Analyzing; **E**: Evaluating; **C**: Creating
R	**U**	**A**	**N**	**E**	**C**	
15%	20%	25%	5%	25%	10 %	

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes (CO):

Sr.	Course Outcome Statements	%weightage
CO-1	Use the syntax and semantics of java programming language and basic concepts of OOP.	**25**
CO-2	Develop reusable programs using the concepts of inheritance, polymorphism, interfaces, collection framework, generic programming and packages.	**30**
CO-3	Demonstrate understanding and use of different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.	**25**
CO-4	Design event driven GUI, networking and IO applications which mimic the real word scenarios.	**20**

Curriculum Revision:	
Version:	1.0
Drafted on (Month-Year):	June-2020
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	June-2025