LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Robotics and Artificial Intelligence (706)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**117062292**
Course Name:	**Computer Programming using Java-2**
Category of Course:	Engineering Science Course (ESC)
Prerequisite Course:	Computer Programming using Java-1 (117062191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
5	**0**	**2**	**6**	**30**

Syllabus				
Unit No.	**Topic**	**Prerequisite Topic**	**Successive Topic**	**Teaching Hours**
01	**Abstraction & Interface with Runtime Polymorphism**			**2 (7%)**
	1.1 Dynamic method dispatch, Object casting and instance of operator	Inheritance (117062191 – Unit - 10)	---	
	1.2 Abstract class, Abstract Method		---	
	1.3 Interface: Implementation of Interface(Partial & Full) Extend Interface		---	
02	**Introduction to Package**			**2 (7%)**
	2.1 Use of Package, Import statement, Access Modifiers	Inheritance (117062191 – Unit - 10)	---	
	2.2 Access control with example		---	
03	**Exception handling**			**3 (10%)**
	3.1 Types of Errors, checked and unchecked Exception	Abstraction & Interface (117062292 – Unit - 01)	---	
	3.2 Exception handling mechanism: Use of try, catch, throw, throws and finally		---	
	3.3 Built in Exception, Custom Exception		---	
04	**Multithreaded Programming**			**4 (13%)**
	4.1 Introduction about Thread	Abstraction & Interface (117062292 – Unit - 01) Exception Handling (117062292 – Unit - 03)	---	
	4.2 Thread life cycle		---	
	4.3 Thread class and Runnable interface		---	
	4.4 Thread methods: start(), run(), getName(), setName(), sleep(), join(), isAlive(), wait(), notify(), currentThread()		---	
	4.5 Synchronized Methods and Synchronized Blocks, Producer – Consumer Problem solution using wait() & notify()		---	
05	**Collection-Part 1**			**3 (10%)**
	5.1 What is collection and need of it. Collection framework Hierarchy, Classes and interfaces in collections, Methods of Collection interface: add(), addAlll(), clear(), contains(), isEmpty(), iterator(), remove(), removeAll(), toArray().	---	---	
	5.2 **List Interface:** **ArrayList**: add(int index, E element), add(E e), clear(), ensureCapacity(int requiredCapacity), get(), set(), isEmpty(), lastIndexOf(Object o), remove(int index), sort(),size(), for each loop to print all elements indexOf(), lastIndexOf() Add Multiple element with Arrays.asList() in Constructor. **Linked List:** Linked list Creation, add(),add(int index, E element), addAll(), addFirst(), addLast(), clear(), contains(), getFirst(), getLast(), remove(), remove(int index), user iterator to print linked list elements **Collections class :** frequency(), reverse(), max(), min(), sort(), Comparator.comparing() to sort class objects by properties.	---	---	
	5.3 **List Interface** **Vector**: create vector, add(), add(index,element), capacity(), clear(), clone(), elementAt(int index), equals(Object o), isEmpty(), remove() **Stack**: Creating a Stack, push(E item), pop(),peek(), search(Object o), empty(), fetch value using iterator()	---	---	

06	**Collection-Part 2**			**3** **(10%)**
	6.1 Queue Interface: **Priority Queue** class : add(object), offer(object), remove(), poll(), element(), peek(): Print Queue using iterator Dequeue Interface: **ArrayDeque** class: ArrayDeque creation, add(Element e), addAll(Collection<? extends E> c), addFirst(Element e), addLast(Element e), clear(), getFirst(), getLast(), isEmpty(), offerFirst(Element e), offerLast(Element e), peek(), remove(), removeFirst(), removeLast(), size()	---	---	
	6.2 Set Interface: **HashSet**: add(), clear(), remove(), isEmpty(), size(), removeAll(), addAll(), equals(), print using iterator Map Interface: **HashMap**: create HashMap, size(), isEmpty(), remove(), put(), putAll(), getKey(), getValue(),print using iterator or foreach **HashTable:** put(), remove(), containsKey(), clear(),getKey(), getValue(),print using iterator or foreach	---	---	
07	**IO Programming**			**3** **(10%)**
	7.1 Introduction to Stream, Byte Stream, Character stream	Constructor (117062191 – Unit - 08)	---	
	7.2 File Class and its method, constructor of File Class, methods like : canExecute(), canRead(), createNewFile(), equals(), exists(), getAbsolutePath(), getName(), getParent(), getParentFile(), getPath(), isDirectory(), isFile(), length(), listFiles(), mkdir(), list().			
	7.3 File Input Stream, File Output Stream			
08	**Character Stream**			**3** **(10%)**
	8.1 Readers and Writers class, FileReader, FileWriter	IO Programming (117062292 – Unit - 07)	---	
	8.2 Buffered Reader, InputStreamReader,			
	8.3 RandomAccessFile with constructor and methods like: close(), readInt(), readUTF(), seek(), writeDouble(), writeFloat(), write(), read(), length(), getFilePointer()			
09	**JDBC Part-1**			**4** **(13%)**
	9.1 JDBC Architecture	----	---	
	9.2 JDBC Drivers, Steps to connect to Database, Connectivity with MySQL, DriverManager, Connection	Abstraction & Interface (117062292 – Unit - 01) Exception Handling (117062292 – Unit - 03)	---	
	9.3 Types of JDBC statements: Statement, Prepared statement, Callable statement	Iterator (117062292 – Unit - 05)	---	
10	**JDBC Part-2**			**3** **(10%)**
	10.1 Database Metadata, Resultset Metadata	JDBC Part-1 (117062292 – Unit - 09) File, File Handling (117062292 – Unit – 07 & 08)	---	
	10.2 Storing image, Retrieving image, Storing file, Retrieving file, Stored procedures, and functions, Transaction Management(commit(). rollback(), setAutoCommit())		---	

Sr No.	Practical Title	Link to Theory Syllabus
1	Write an application that generates custom exception if any value from its command line arguments is negative.	Unit – 3
2	Write a method for computing xy by doing repetitive multiplication. x and y are of type integer and are to be given as command line arguments. Raise and handle exception(s) for invalid values of x and y. Also define method main. Use finally in above program and explain its usage.	Unit – 3
3	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 2followings:free: indicated by integer value 0 occupied: indicated by integer value 1 inaccessible: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from command line arguments and initialize the array. Raise and handle user defined exception if invalid status value given	Unit – 3

4	Write a complete program to accept N integer numbers from the command line. Raise and handle exceptions for following cases : - when a number is –ve - when a number is evenly divisible by 10 - when a number is greater than 1000 and less than 2000 - when a number is greater than 7000 Skip the number if an exception is raised for it, otherwise add it to find total sum	Unit – 3
5	Declare a class called book having author_name as private data member. Extend book class to have two sub classes called book_publication&paper_publication. Each of these classes have private member called title. Write a complete program to show usage of dynamic method dispatch (dynamic polymorphism) to display book or paper publications of given author. Use command line arguments for inputting data. .	Unit – 1
6	Write a program that reads file name from user, through command line argument and displays/reads content of the text file on console.	Unit – 6
7	Write a program that reads file name from user, through command line argument and displays/reads content of the text file on console.	Unit – 6
8	Write a program to replace all “word1” by “word2” from a file1, and output is written to file2 file and display the no. of replacement.	Unit – 6
9	Write a program that counts the no. of words in a text file. The file name is passed as a command line argument. The program should check whether the file exists or not. The words in the file are separated by white space characters.	Unit – 7
10	Write a program to read the content of a file into a character array and write it into another file. Get names of the files from command line	Unit – 7
11	Read employee salary and calculate the income tax based on 10% of income and store it in tax.txt file for five different employees	Unit – 7
12	The abstract Vegetable class has three subclasses named Potato, Brinjal and Tomato. Write an application that demonstrates how to establish this class hierarchy. Declare one instance variable of type String that indicates the color of a vegetable. Create and display instances of these objects. Override the toString() method of Object to return a string with the name of the vegetable and its color	Unit – 1
13	Write a program that illustrates interface inheritance. Interface P is extended by P1 And P2. Interface P12 inherits from both P1 and P2.Each interface declares one constant and one method. Class Q implements P12.Instantiate Q and invokes each of its methods. Each method displays one of the constants	Unit – 1
14	The Transport interface declares a deliver() method. The abstract class Animal is the superclass of the Tiger, Camel, Deer and Donkey classes. The Transport interface is implemented by the Camel and Donkey classes. Write a test program that initialize an array of four Animal objects. If the object implements the Transport interface, the deliver() method is invoked. .	Unit – 1
15	Write a abstract class named Person and its two subclasses named student and Employee. A person has a name, address, phone number and email address. A student has enrollment course. An Employee has an office, salary, and designation. Define constructors and methods for input and display for both classes. Define constructor and methods for input and display for both classes. Write a main program to give demonstration of all.	Unit – 1
16	Write a complete multi-threaded program to meet following requirements: - Read matrix [A] m x n - Create m number of threads - Each thread computes summation of elements of one row, i.e. ith row of the matrix is processed by ith thread. Where 0 <= i< m. - Print the results	Unit – 4
17	Write an application that executes two threads. One thread displays "Good Morning" every 1000 milliseconds & another thread displays "Good Afternoon" every 3000 milliseconds. Create the threads by implementing the Runnable interface.	Unit – 4
18	Write a complete multi-threaded program to meet following requirements: o Two threads of same type are to be instantiated in the method main. o Each thread acts as a producer as well as a consumer. o A shared buffer can store only one integer information along with the source & destination of the information at a time. o The information produced is to be consumed by appropriate consumer. o Both producers produce information for both consumers. o Each thread produces 5 information	Unit – 4
19	Write a multithreaded program to print all odd positive numbers in ascending order up to n, where n is a positive integer number given as a command line argument. Instantiate requited number of threads, where each thread except the last, examines next 50 numbers and the last thread examines remaining numbers up to n.	Unit – 4
20	Write a complete multi threaded program to meet following requirements for producerconsumer threads: - Three threads – one producer and two consumers to be instantiated in the method main. - At a time, the producer produces one integer information along with consumer_id to represent id of a consumer that will consume produced information. - Information and consumer_id are stored in a shared buffer. - The information produced is to be consumed by appropriate consumer only, as specified by the producer. - The producer thread produces total 6 information	Unit – 4
21	Write a complete multi threaded program to meet following requirements for producerconsumer threads: - Three threads – one producer and two consumers to be instantiated in the method main. - At a time, the producer produces one integer information along with consumer_id to represent id of a consumer that will consume produced information. - Information and consumer_id are stored in a shared buffer. - The information produced is to be consumed by appropriate consumer only, as specified by the producer. - The producer thread produces total 6 information	Unit – 4

22	Consider Bank table with attributes AccountNo,CustomerName, Balance, Phone and Address. Write a database application which allows insertion, updation and deletion of records in Bank table. Print values of all customers whose balance is greater than 20,000.	Unit – 10
23	Write a program using JDBC for getting personal information – name, birthdate, sex, address, phone no, email-id & store it in database. Also provide list of all records, all male, all female & all minors (age below 18).	Unit – 10
24	Write Java application program to change the basic = basic + 500 of all the employees whose age is greater then 40 from employee table then display how many record updated	Unit – 10
25	Give the use of Statement, PreparedStatement and CallableStatement object and Write code to insert three records into student table using PreparedStatement (assume student table with Name, RollNo, and Branch field).	Unit – 9

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	JDK, JRE, VS CODE, PhpMyAdmin, My SQL

Proposed Theory + Practical Evaluation Scheme by Academicians
(% Weightage Category Wise and it's Marks Distribution)

L :	5	T:	0	P:	2

Note : In Theory Group, Total 4 Test (T1+T2+T3+T4) will be conducted for each subject.
Each Test will be of 25 Marks.
Each Test Syllabus Weightage: Range should be 20% - 30%

Group (Theory or Practical)	Group (Theory or Practical) Credit	Total Subject Credit	Category	% Weightage	Marks Weightage
Theory	5	6	MCQ	20%	40
Theory			Theory Descriptive (Mainly Programming)	30%	60
Theory			Formulas and Derivation	0%	0
Theory			Numerical	0%	0
Expected Theory %	**50%**		**Calculated Theory %**	**50%**	**100**
Practical	1		Individual Project	25%	50
Practical			Group Project	15%	30
Practical			Internal Practical Evaluation (IPE)	10%	20
Practical			Viva	0%	0
Practical			Seminar	0%	0
Expected Practical %	**50%**		**Calculated Practical %**	**50%**	**100**
Overall %	**100%**			**100%**	**200**

Course Outcome	
	Upon completion of the course students will be able to
CO1	Apply concept of abstraction, interface, packages and Exception handling to create error free code.
CO2	Implement multithreading in object programs, understand use of collection (ArrayList, LinkedList, Vector and Stack) in programs.
CO3	Understand different kind of file I/O programming, use of collection (Queue, set, Hash Map and Hash Table) in programs.
CO4	Apply the concepts of JDBC, Transection processing, statement objects and Resultset to perform operations on Database.
Suggested Reference Books	
1	Java: The Complete Reference, Tenth Edition (Complete Reference Series), Herbert Schildt – McGrawHill
2	Java Server Programming Java EE 7 (J2EE 1.7), Black Book Kindle Edition, Kogent Learning Solutions Inc – Dreamtech
3	Core Java Volume I--Fundamentals, 11th edition, Cay Horstman – Pearson
4	Core Java - An Integrated Approach Includes All Version Upto Java 8, Dr. R. Nageswara Rao – Dreamtech
5	Programming with Java by, E Balagurusamy – McGrawHill

List of Open Source Software/Learning website	
1	https://www.javatpoint.com/java-tutorial
2	https://www.tutorialspoint.com/java/index.htm
3	https://www.geeksforgeeks.org/java/
4	https://www.oracle.com/java/technologies/downloads/#jdk17-windows & https://notepad-plus-plus.org/downloads/
5	https://www.programiz.com/java-programming/