ITE2005	Advanced Java Programming	L	T	P	J	C
		3	0	2	0	4
Pre-requisite	**ITE1002**	**Syllabus version**				
		1.1				

Course Objectives:

- To apply the core Java fundamentals to learn the advanced concepts in J2SE
- To design and develop web application development and database connectivity using Servlets, JSP and JDBC
- To apply the advanced Java frameworks for improving the web application design

Expected Course Outcome:

1) Provide a basic understanding of core Java concepts.
2) Comprehend Java's support in parallel programming, GUI creation and network programming.
3) Design and develop server side programming using Servlets.
4) Develop web applications using JSP.
5) Understand and implement MVC architecture with Struts framework.
6) Use JSF framework to build better user interfaces.
7) Integrate Hibernate framework with applications for Object Relational Mapping.
8) Design and develop advanced enterprise web applications and rich internet applications

Student Learning Outcomes (SLO):	**2, 5, 17**
[2]	Having a clear understanding of the subject related concepts and of contemporary issues
[5]	Having design thinking capability
[17]	Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice

Module:1	**Introduction to Java Programming:**	**6 hours**

Features of Java, Data Types, Variables, Operators, Arrays, Control Statements. Introducing Classes and Objects, Methods, Inheritance, Packages and Interfaces, Exception Handling, Inner classes, String Handling

Module:2	**Exploring Core Java**	**6 hours**

Multithreaded Programming, Files and IO Streams, Object Serialization ,Applets , Java GUI Programming and Event Handling, Java Networking, RMI, Reflection, Collections, Generics, Java Auto boxing and Annotations

Module:3	**Introducing JavaEE**	**6 hours**

Enterprise Java, Basic Application Structure, Using Web Containers, Creating Servlets, Configuring Servlets, Understanding HTTP methods, Using Parameters and Accepting Form Submissions, Using Init parameters, File Uploading, JDBC

Module:4	**Java Server Pages**	**6 hours**
Creating JSPs, Using Java within JSP, Combining Servlets and JSPs, Maintaining State using Sessions, JSP 2.0 EL, Using Javabeans components in JSP Documents, JSP Custom Tag Library, Integrating Servlets and JSP: Model View Controller Architecture		
Module:5	**Struts Framework**	**6 hours**
Introduction to Struts – Building a Simple Struts Application – Understanding Model, View and Controller Layer- Overview of Tiles		
Module:6	**Java Server Faces(JSF)**	**7 hours**
Introduction to Java Server Faces (JSF)- JSF Application Architecture – Building a simple JSF Application - JSF Request Processing Lifecycle – The Facelets View Declaration Language – User Interface Component Model- JSF Event Model		
Module:7	**Spring Framework and Hibernate**	**6 hours**
Understanding Inversion of Control (IoC), Aspect Oriented Programming (AOP) and Dependency Injection, MVC pattern for Web Applications, Spring Framework, Understanding Application Context, Bootstrapping Spring framework, Configuring Spring framework, Data Persistence, Object/relational Mapping, Hibernate ORM, Mapping Entities to Tables		
Module:8	**Contemporary issues:**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Book(s)

1. Herbert Schildt, The Complete Reference-Java, Tata Mcgraw-Hill Edition, Eighth Edition, 2014.

Reference Books

1. Nicholas S. Williams, Professional Java for Web Applications, Wrox Press, 2014.
2. Ed Burns, Chris Schalk, JavaServer Faces 2.0, The Complete Reference, McGraw-Hill Publishers, 2010.
3. Christian Bauer, Gavin King, Gary Gregory, Java Persistence with Hibernate, 2015.
4. Craig Walls, Spring in Action Paperback, Manning Publications, 2014.

List of Challenging Experiments (Indicative)

1. Write a program to read the First name and Last name of a person, his weight and height using command line arguments. Calculate the BMI Index which is defined as the individual's body mass divided by the square of their height.

Category	BMI Range-Kg/m^2
Underweight	<18.5
Normal (healthy weight)	18.5 to 25
Overweight	25 to 30
Obese Class	Over 30

Display the name and display his category based on the BMI value thus calculated.

2.	If there are 4 batches in B.Tech (IT) learning 'ITE101' course, read the count of the slow learners (who have scored <25) in each batch. Tutors should be assigned in the ratio of 1:4 (For every 4 slow learners, there should be one tutor). Determine the number of tutors for each batch. Create a 2-D jagged array with 4 rows to store the count of slow learners in the 4 batches. The number of columns in each row should be equal to the number of groups formed for that particular batch (Eg., If there are 23 slow learners in a batch, then there should be 6 tutors and in the jagged array, the corresponding row should store 4, 4, 4, 4, 4,3). Use for-each loop to traverse the array and print the details. Also print the number of batches in which all tutors have exactly 4 students.
3.	Write a program to read a chemical equation and find out the count of the reactants and the products. Also display the count of the number of molecules of each reactant and product. Eg., For the equation, 2NaOH + H2SO4 -> Na2SO4+ 2H2O, the O/P should be as follows. Reactants are 2 moles of NaOH, 1 mole of H2SO4. Products are 1 mole of Na2SO4 and 2 moles of H2O.
4.	(Bioinformatics: finding genes) Biologists use a sequence of letters A, C, T, and G to model a genome. A gene is a substring of a genome that starts after a triplet ATG and ends before a triplet TAG, TAA, or TGA. Furthermore, the length of a gene string is a multiple of 3 and the gene does not contain any of the triplets ATG, TAG, TAA, and TGA. Write a program that prompts the user to enter a genome and displays all genes in the genome. If no gene is found in the input sequence, displays no gene. Here are the sample runs: Enter a genome string: TTATGTTTTAAGGATGGGGCGTTAGTT O/P: TTT GGGCGT
5.	Create a class Film with string objects which stores name, language and lead_actor and category (action/drama/fiction/comedy). Also include an integer data member that stores the duration of the film. Include parameterized constructor, default constructor and accessory functions to film class. Flim objects can be initialized either using a constructor or accessor functions. Create a class Film Main that includes a main function. In the main function create a vector object that stores the information about the film as objects. use the suitable methods of vector class to iterate the vector object to display the following a. The English film(s) that has Arnold as its lead actor and that runs for shortest duration. b. The Tamil film(s) with Rajini as lead actor. c. All the comedy movies.
6.	Define an abstract class 'Themepark' and inherit 2 classes 'Queensland' and 'Veegaland' from the abstract class. In both the theme parks, the entrance fee for adults is Rs.500 and for children it is Rs.300. If a family buys 'n' adult tickets and 'm' children tickets, define a method in the abstract class to calculate the total cost. Also, declare an abstract method playGame() which must be redefined in the subclasses. In Queensland, there are a total of 30 games. Hence create a Boolean array named 'Games' of size 30 which initially stores false values for all the elements. If the player enters any game code that has already been played, a warning message should be displayed and the user

	should be asked for another choice. In Veegaland, there are a total of 40 different games. Thus create an integer array with 40 elements. Here, the games can be replayed, until the user wants to quit. Finally display the total count of games that were repeated and count of the games which were not played at all.
7.	Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an Illegal Argument Exception. If the Mobile Number contains any character other than a digit, raise a Number Format Exception. If the Register Number contains any character other than digits and alphabets, throw a No Such Element Exception. If they are valid, print the message 'valid' else 'invalid'
8.	Within the package named 'primes package', define a class Primes which includes a method checkForPrime() for checking if the given number is prime or not. Define another class named Twin Primes outside of this package which will display all the pairs of prime numbers whose difference is 2. (Eg, within the range 1 to 10, all possible twin prime numbers are (3,5), (5,7)). The TwinPrimes class should make use of the checkForPrime() method in the Primes class.
9.	Define a class 'Donor' to store the below mentioned details of a blood donor. - Name, age, Address, Contact number, bloodgroup, date of last donation. Create 'n' objects of this class for all the regular donors at Vellore. Write these objects to a file. Read these objects from the file and display only those donors' details whose blood group is 'A+ve' and had not donated for the recent six months.
10.	Three students A, B and C of B.Tech-IT II year contest for the PR election. With the total strength of 240 students in II year, simulate the vote casting by generating 240 random numbers (1 for student A, 2 for B and 3 for C) and store them in an array. Create four threads to equally share the task of counting the number of votes cast for all the three candidates. Use synchronized method or synchronized block to update the three count variables. The main thread should receive the final vote count for all three contestants and hence decide the PR based on the values received.
11	Draw a ball, filled with default color. Move the ball from top to bottom of the window continuously with its color changed for every one second. The new color of the ball for the next second should be obtained by adding 20 to the current value of Red component, for the second time by adding 20 to the blue component, and for the third time by adding 20 to the blue component, till all reach the final limit 225, after which the process should be repeated with the default color.
12.	Develop a UDP based client-server application to notify the client about the integrity of data sent from its side. Check sum calculation: 1. Add the 16-bit values up. Each time a carry-out (17th bit) is produced, swing that bit around and add it back into the LSb (one's digit). 2. Once all the values are added in this manner, invert all the bits in the result. For example, separate the data into groups of 4 bits only for readability. 1000 0110 0101 1110 1010 1100 0110 0000

	0111 0001 0010 1010 First, add the 16-bit values 2 at a time: 1000 0110 0101 1110 First 16-bit value + 1010 1100 0110 0000 Second 16-bit value --- 1 0011 0010 1011 1110 Produced a carry-out, which gets added + \----------> 1 back into LBb --- 0011 0010 1011 1111 + 0111 0001 0010 1010 Third 16-bit value --- 0 1010 0011 1110 1001 No carry to swing around (**) --- 0010 0101 1001 1111 Our "one's complement sum" Then take the one's complement of the sum which is 1101 1010 0110 0000 The "one's complement" So the checksum stored in the header should be 1101 1010 0110 0000.
13.	Develop an RMI application to invoke a remote method that takes two numbers and returns true if one number is an exact multiple of the other and false otherwise. Eg., 5 and 25 -> true 26 and 13 -> true 4 and 18 -> false
14.	a)Assume two cookies are created whenever a VIT student visits the VIT webpage-one for his/her name and the other for his campus. For subsequent visits, he/she should be greeted with the message similar to the one below “Hi Ajay from Chennai Campus!!”. Write a servlet program to do the needful. b)Build an application using JSF framework to implement a Celsius to Fahrenheit converter. Note: Fahrenheit=(Celsius*9/5)+32
15.	Using Hibernate framework, simulate the course registration process for Advanced Java Programming. Let the registration number and name of the students who register for the course, be stored in a database. The tool should allow deletion of the registered course for a particular student, if he/she wishes. At any instant, the list of students who have registered for the course should be displayed, if requested for.

Total Laboratory Hours	30 hours

Recommended by Board of Studies	12-08-2017		
Approved by Academic Council	No. 47	Date	05-10-2017