

**PESIT Bangalore South Campus**  
**Department of Computer Science and Engineering**

<b>ADVANCED JAVA AND J2EE</b> [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2017) SEMESTER - V			
Subject Code	15CS553	IA Marks	20
Number of Lecture /Week	4	Exam Marks	80

**Faculty: Prof. Sajeevan K / Prof. Lakshmi Nagaprasanna**

**No.of Sessions: 56**

**Course Description:**

This *Advanced Java* course aimed towards the Students who already has learned the *Fundamentals of Java Programming*. In this *Advanced Java Programming* course, expert builds on the beginners Java course, and goes deeper into programming topics that help you to understand these more advanced Java concepts. Some of the advanced topics that will cover in this Advanced Java courses includes; Enumerations, Autoboxing and Annotations(metadata), sequential and associative data structures, The collections and Framework, Networking, database programming with JDBC, Java Beans and Java Servlets.

**Prerequisites:**

Knowledge in Core Java (up to swing package)

**Course objectives:** This course will enable students to

- Identify the need for advanced Java concepts like Enumerations and Collections
- Demonstrate the use of JavaBeans to develop component-based Java software
- Adapt Servlets to build server side programs
- Make use of JDBC to access database through Java Programs
- Construct client-server applications using Java socket API
- Construct an enterprise application using Entity Beans linked with Database
- Construct an asynchronous enterprise application using Message-Driven Beans
- Map java inheritance hierarchy with database tables using various mapping techniques.
- Persist different types of collections.

**Course Plan:**

Class	Chapter Title	Topics to be covered	Book	% of portion covered	
					Cumulative
1.	Module I	<b>Enumerations, Autoboxing and Annotations(metadata):</b> Enumerations, Enumeration fundamentals	T2:1.1	20%	20%
2.		the values() and valueOf() Methods	T2:1.2		
3.		Java enumerations are class types	T1:2.1		
4.		enumerations Inherits Enum, example	T2:1.3		
5.		Type wrappers, Autoboxing	T1:2.2		
6.		Autoboxing and Methods,	T1:2.3		
7.		Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing	T1:2.4		
8.		Boolean and character values , Autoboxing/Unboxing helps prevent errors	T1:1.3		
9.		A word of Warning. Annotations, Annotation basics, specifying retention policy	T1:1.3		
10.		Obtaining Annotations at run time by use of reflection, Annotated element Interface	T1:1.4		
11.		Using Default values, Marker Annotations, Single Member annotations, Built-In annotations	T1:1.4		
12.	Module II	<b>The collections and Framework:</b> Collections Overview	T2:3.1	20%	40%
13.		Recent Changes to Collections, The Collection Interfaces	T2:3.1		
14.		The Collection Classes	T2:3.3		
15.		Accessing a collection Via an Iterator	T2:3.4		
16.		Storing User Defined Classes in Collections	T1:4.1		
17.		The Random Access Interface, Working With Maps	T1:4.2		
18.		Comparators, The Collection Algorithms	T2:3.8		
19.		Why Generic Collections? The legacy Classes and Interfaces	T1:5.3		
20.		Parting Thoughts on Collections	T1:5.3		
21.	Module III	<b>Networking:</b> Introduction, Networking Basics	T2:4.1	20%	60%
22.		Protocols, Internet Address (IPv4 and IPv6)	T2:4.3		

23.		Ports, Classes and Interfaces in java.net	T2:4.5		
24.		class InetAddress, Constructors of class in Serverside Program InetAddress	T1:9.1		
25		Methods of classInetAddress, class URL, Constructors of URL class	T1:9.1		
26.		Methods of URL class, class URL Connection	T1:9.2		
27.		TCP/IP Server Socket Programming,	T1:9.2		
28.		Constructors of classSocket, Methods of class Socket, Class ServerSocket	T1:9.3		
29.		Constructors of class ServerSocket, Methods of class ServerSocket	T1:9.3		
30.		Communication through Sockets, Client-side Socketp programming	T1:9.4		
31.		Users/UnreliableDatagramProtocol, classDatagramPacket	T1:6.4.		
32.		Constructors, Methods of class Datagram Packets, DatagramSocket Class, Constructors of class DatagramSocket, Methods of class DatagramSocket	T1:6.4.		
33.		<b>Java Beans:</b> Introduction to Java Beans	T2:5.1		
34.		Attributes of Beans, Benefits of Using Beans, Properties of a Bean	T2:5.2		
35.		Java Bean API, Interfaces, Classes of java. Bean package(Java SE 8),class Beans	T1:8.2		
36.		Class Methods, class Property Change Support, JAR Files	T1:8.2		
37.		Creating a JAR file, Viewing the contents of a JAR File	T1:8.3		
38.		Building Java Beans with NETBEAN IDE, Building a Composite Component Bean	T1:8.4		
39.		Changing Properties of Components, Source Code Generated by IDE	T1:8.4		
40.	<b>Module IV</b>	Java Beans Project 1—Construct a Bean Containing a Label and a Scrollbar, Java Beans Project 2— Construct a Bean to Illustrate MouseEvents	T2:5.4	<b>20%</b>	<b>80%</b>
41.		Java Beans Project 3— Construct a Bean to Illustrate Working of RadioButtons, Java Beans Project 4—Construct a Bean Containing ComboBox, Java Beans Project 5,Application Program, Application Project 1,Application Project 2	T2:5.4		

42.		<b>Java Servlets:</b> Introduction, HTTP Basics, Life Cycle of a Servlet, Servlet Architecture, Setting up Web Server for Servlet Deployment, Classes Defined in Javax.servlet.http Package	T2:5.9			
43.		Interfaces Defined in javax.servlet.http Package, Handling HTTP Request and Response, Writing Servlet, Reading Servlet Parameters, Classes in javax.servlet Package, Session Tracking and Session Management, Session Tracking Techniques, Http Redirects in Servlets, Securing Servlets, Application Program	T2:5.8			
44.	<b>Module V</b>	<b>Java Database Connectivity:</b> Introduction	T2:7.1	<b>20%</b>	<b>100%</b>	
45.		JDBC Architecture	T1:12.1			
46.		Two-tier Architecture for DataAccess, Three-tier Architecture for DataAccess	T1:12.1			
47.		Installing MySQL and MySQL Connector/J,SQL Statements	T2:7.4			
48.		JDBC Environment Setup, JDBC Connectivity Model and API	T2:7.5			
49.		Establishing JDBC Database Connections	T1:12.2			
50.		Load and Register the JDBC Driver	T1:12.2			
51.		Defining the ConnectionURL	T2:8.2, T1:12.2			
52.		ResultSet Interface	T2:8.2			
53.		Navigating the ResultSet	T2:11.1			
54.		ResultSetMetaData interface	T2:11.1			
55.		Creating JDBC Application	T2:11.1			
56.		JDBC Batch Processing, JDBC Transaction Management, Application Programs.				

### Course Outcomes:

- Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs
- Build client-server applications and TCP/IP socket programs
- Develop reusable software components using Java Beans
- Describe how servlets fit into Java-based web application architecture
- Illustrate database access and details for managing information using the JDBC API

## **Text Books:**

### **Text Books:**

1. Herbert Schildt: Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.
2. Uttam K. Roy , Advanced JAVA Programming. Oxford University Press.

### **Reference Book:**

1. Jim Keogh: J2EE - The Complete Reference, Tata McGraw Hill, 2007.
2. Y. Daniel Liang: Introduction to JAVA Programming, 7th Edition, Pearson Education, 2007.
3. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education, 2004.java, Tata McGraw Hill education private limited.
5. Richard A Johnson, Introduction to Java Programming and OOAD, CENGAGE Learning.
6. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.

## **QUESTION BANK:**

- 1) What is JDBC?
- 2) What is JDBC Driver?
- 3) What are the steps to connect to the database in java?
- 4) What are the JDBC API components?
- 5) What are the JDBC statements?
- 6) What is the difference between Statement and PreparedStatement interface?
- 7) How can we execute stored procedures and functions?
- 8) What is the role of JDBC DriverManager class?
- 9) What does the JDBC Connection interface?
- 10) What does the JDBC ResultSet interface?
- 11) What does the JDBC ResultSetMetaData interface?
- 12) What does the JDBC DatabaseMetaData interface?
- 13) Which interface is responsible for transaction management in JDBC?
- 14) What is batch processing and how to perform batch processing in JDBC?
- 15) How can we store and retrieve images from the database?
- 16) What is the difference between ArrayList and Vector?
  
- 17) What is the difference between ArrayList and LinkedList?
  
- 18) What is the difference between Iterator and ListIterator?
  
- 19) What is the difference between Iterator and Enumeration?

- 20) What is the difference between List and Set?
- 21) What is the difference between HashSet and TreeSet?
- 22) What is the difference between Set and Map?
- 23) What is the difference between HashSet and HashMap?
- 24) What is the difference between HashMap and TreeMap?
- 25) What is the difference between HashMap and Hashtable?
- 26) What is the difference between Collection and Collections?
- 27) What is the difference between Comparable and Comparator?
- 28) What is the advantage of Properties file?
- 29) What does the hashCode() method?
- 30) Why we override equals() method?
- 31) How to synchronize List, Set and Map elements?
- 32) What is the advantage of generic collection?
- 33) What is hash-collision in Hashtable and how it is handled in Java?
- 34) What is the Dictionary class?
- 35) What is the default size of load factor in hashing based collection?
- 36) What is the difference between Array and ArrayList?
- 37) What is the difference between length of Array and size of ArrayList?
- 38) How to convert ArrayList to Array and Array to ArrayList?
- 39) How to make Java ArrayList Read-Only?
- 40) How to remove duplicates from ArrayList?

- 41) How to reverse ArrayList?
- 42) How to sort ArrayList in descending order?
- 43) How to synchronize ArrayList?