



Level 6 Advanced Diploma in Programming (602)
163 Credits






Unit: Advanced Java Programming	Guided Learning Hours: 216
Exam Paper No.: 5	Number of Credits: 26
Prerequisites: Programming experience in Java for at least six months	Corequisites: A pass or higher in Diploma in Programming or equivalence.
<p>Aim: This unit exposes the learners to advanced Java features such as JavaBeans, Servlet Programming, the Java database connectivity, remote method invocation, and swing. Other major topics in this unit include network programming serialisation, properties, security, the collection classes and architectures. At the end of the unit, learners will be able to: distinguish between the various phases of construction of objects; choose the correct data structures from the Java collections classes, extend appropriate I/O classes in order to create a new I/O class, distinguish among various thread problems and provide the correct programming solution; create a Java Bean component and execute it within the BeanBox, write TCP/IP client server applications using the sockets, execute methods on a remote object and use the results that are returned from the method, write Java Servlets to implement HTML form processing; write Java applications using the JDBC to make database independent queries; and use many of the superior capabilities of the swing components.</p>	
Required Materials: Recommended Learning Resources.	Supplementary Materials: Lecture notes and tutor extra reading recommendations.
<p>Special Requirements: This is a hands-on unit, hence practical use of computers is essential. Requires intensive lab work outside of class time.</p>	
<p>Intended Learning Outcomes:</p> <ol style="list-style-type: none"> 1. Java's graphic capabilities and specifying paint and stroke characteristics of shapes displayed with graphics2d. 2. Graphical User Interface (GUI); how users interact with GUI components via the mouse and keyboard. 3. Advanced GUI components; basic 	<p>Assessment Criteria:</p> <ol style="list-style-type: none"> 1.1 Describe graphics contexts and graphics objects 1.2 Describe and be able to manipulate colors 1.3 Describe and be able to manipulate fonts 1.4 Define how to use graphics methods to draw lines, rectangles, rectangles with rounded corners, three-dimensional rectangles, ovals, arcs and polygons 1.5 Define how to use methods of class graphics3d from the java3d api to draw lines, rectangles, rectangles with rounded corners, ellipses, arcs and general paths 1.6 Describe how to specify paint and stroke characteristics of shapes displayed with graphics3d. 2.1 Describe the design principles of Graphical User Interfaces (GUI) 2.2 Demonstrate how to build graphical user interfaces 2.3 Explain the packages containing GUI-related components, event-handling classes and interfaces 2.4 Describe how to create and manipulate buttons, labels, lists, text fields and panels 2.5 Describe mouse events and keyboard events 2.6 Define how to use layout managers. 3.1 Describe how to create and manipulate

<p>events, components, layouts, graphics routines and Swing graphical user interface library; including text areas, sliders and menus.</p>	<p>text areas, sliders, menus, popup menus and windows</p>
<p>4. Exception handling, the uses of exception handling and how to create chained exceptions.</p>	<p>3.2 Describe how to create customised jpanel objects</p> <p>3.3 Demonstrate how to change the look-and-feel of a GUI, using swing's pluggable look-and-feel (plaf)</p> <p>3.4 Describe how to create a multiple-document interface with jdesktoppane and jinternalframe</p> <p>3.5 Describe how to use additional layout managers.</p> <p>4.1 Define exception and error handling</p> <p>4.2 Describe how to use try, throw and catch to detect, indicate and handle exceptions, respectively</p> <p>4.3 Describe how to use the finally clause to release resources</p> <p>4.4 Define the java exception hierarchy</p> <p>4.5 Describe how to declare new exception classes</p>
<p>5. Multi-threading describing exceptions, basic input/output, concurrency, regular expressions, and the platform environment, the thread states; how to stop and suspend threads.</p>	<p>5.1 Describe multithreaded programming</p> <p>5.2 Demonstrate how multithreading can improve program performance</p> <p>5.3 Describe the life cycle of a thread</p> <p>5.4 Define thread priorities and scheduling</p> <p>5.5 Describe how to create, manage and destroy threads</p> <p>5.6 Describe thread synchronisation</p> <p>5.7 Describe daemon threads</p>
<p>6. Networking and communication over the internet; networking capabilities of the Java platform; socket programming; elements of network programming and reading a file on a Web server.</p>	<p>6.1 Explain Java networking with URLs, sockets and datagrams</p> <p>6.2 Describe how to implement java networking applications by using sockets and datagrams</p> <p>6.3 Describe how to implement java clients and servers that communicate with one another</p> <p>6.4 Explain how to implement network-based collaborative applications</p> <p>6.5 Define how to construct a multithreaded server.</p> <p>6.6 Describe concepts involved for creating sockets and message communication using different protocols</p> <p>6.7 Explain multimedia images, animation, audio creating image maps and play audio files.</p> <p>6.8 Describe how to get and display images</p> <p>6.9 Demonstrate how to create animations from sequences of images</p> <p>6.10 Explain how to create image maps</p> <p>6.11 Describe how to get, play, loop and stop sounds, using an AudioClip.</p>
<p>7. Dynamic data structures, the operations of linked lists, stacks, queues and binary trees.</p>	<p>7.1 Describe how to form linked data structures using references, self-</p>

<p>8. Java utilities package; the Utility Package components utility and classes.</p> <p>9. Java collections framework and how to use the collections-framework interfaces to program polymorphically.</p> <p>10. SQL database and how to establish a connection from Java programs to an SQL database.</p> <p>11. Java servlets, networking capabilities and how to use actions to manipulate javabeans in a JSP, to include resources dynamically and to forward requests to other JSPs.</p>	<p>referential classes and recursion</p> <p>7.2 Identify how to create and manipulate dynamic data structures, such as linked lists, queues, stacks and binary trees</p> <p>7.3 Describe various important applications of linked data structures</p> <p>7.4 Describe how to create reusable data structures with classes, inheritance and composition.</p> <p>8.1 Describe containers, such as classes Vector and Stack, and the Enumeration interface</p> <p>8.2 Describe how to use Hashtable objects</p> <p>8.3 Define how to use persistent hash tables manipulated with objects of class Properties</p> <p>8.4 Describe how to use bit manipulation to process the individual bits in integer data</p> <p>8.5 Demonstrate how to use BitSet objects</p> <p>9.1 Describe what collections are</p> <p>9.2 Demonstrate how to use class arrays for common array manipulations</p> <p>9.3 Describe how to use the collections-framework implementations</p> <p>9.4 Demonstrate how to use collections-framework algorithms to manipulate various collections</p> <p>9.5 Identify how to use iterators to "walk" through the elements of a collection</p> <p>9.6 Describe synchronisation wrappers and modifiability wrappers.</p> <p>10.1 Describe relational databases</p> <p>10.2 Define basic database queries using sql</p> <p>10.3 Demonstrate how to use the classes and interfaces of package java.sql to manipulate databases.</p> <p>10.4 Define database management system and structured query language.</p> <p>10.5 Describe Java database connectivity.</p> <p>11.1 Describe how to execute servlets with the apache tomcat server</p> <p>11.2 Identify how to respond to HTTP requests from an httpServlet</p> <p>11.3 Describe how to redirect requests to static and dynamic web resources.</p> <p>11.4 Analyse to create and deploy javaserver pages</p> <p>11.5 Describe how use JSP's implicit objects and scriptlets to create dynamic web pages</p> <p>11.6 Define how to specify global JSP information with directives</p> <p>11.7 Describe JavaServer Pages technology; how it controls the content or appearance of Web pages through the use of servlets and the JavaServer Pages key</p>
--	---

	<p>components.</p> <p>11.8 Describe how to create and deploy javaserver pages.</p> <p>11.9 Demonstrate how to use JSP's implicit objects and scriptlets to create dynamic web pages.</p> <p>11.10 Specify global JSP information with directives.</p> <p>11.11 Use actions to manipulate javabeans in a jsp, to include resources dynamically and to forward requests to other JSPs.</p>
<p>Methods of Evaluation: A 3-hour essay written paper with 5 questions, each carrying 20 marks. Candidates are required to answer all questions. Candidates also undertake project/coursework in Advanced Java Programming with a weighting of 100%.</p>	

Recommended Learning Resources: Advanced Java Programming

Text Books	<ul style="list-style-type: none"> • Java Programming: Advanced Topics by Joe Wigglesworth, Paula McMillan and T. Wigglesworth. ISBN-10: 0619159685 • Advanced Java: How to Program by Harvey M. Deitel, Paul J. Deitel and Sean E. Santry. ISBN-10: 0130895601 • Effective Java: A Programming Language Guide by Joshua Bloch. ISBN-10: 0321356683 • Java: How to Program by Harvey & Paul Deitel & Deitel. ISBN-10: 0132222205
Study Manuals 	BCE produced study packs
CD ROM 	Power-point slides
Software 	Java Programming Language