




---

## LONDON CAPITAL COMPUTER COLLEGE

---

### Diploma in Programming (601) - Java

<p><b>Prerequisites:</b> Basic programming skills or basic knowledge of computer use.</p>	<p><b>Corequisites:</b> A pass or higher in Diploma in System Design or equivalence.</p>
<p><b>Aim:</b> This course takes candidates through basic Java syntax, how to design simple programs and classes used in the development of Java applications and applets.</p>	
<p><b>Required Materials:</b> Student study materials</p>	<p><b>Supplementary Materials:</b> Recommended textbooks and lecture notes.</p>
<p><b>Special Requirements:</b> This is a hands-on course, hence practical use of computers is essential. Requires intensive lab work outside of class time.</p>	
<p><b>Intended Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Discover the history of Java. Identify the basics of the Java environment. Define object-oriented programming.</li>   <li>2. Introduce simple examples to illustrate Java features. Define Java applications and primitive data types.</li>   <li>3. Define Java Development kit. Discuss Java Applets. Define how to draw strings and lines.</li>   <li>4. Describe the structured-programming principle. Define how control structures help</li> </ol>	<p><b>Assessment Criteria:</b></p> <ol style="list-style-type: none"> <li>1.1 Familiarise with different types of programming languages</li> <li>1.2 Describe a typical java development environment</li> <li>1.3 Describe java's role in developing distributed client/server applications for the internet and web</li> <li>1.4 Define object-oriented design with the UML and design patterns.</li>   <li>2.1 Describe how to write simple java applications</li> <li>2.2 Demonstrate how to use input and output statements</li> <li>2.3 Familiarise with primitive types</li> <li>2.4 Describe basic memory concepts</li> <li>2.5 Describe how to use arithmetic operators</li> <li>2.6 Describe arithmetic-operator precedence</li> <li>2.7 Demonstrate how to write decision-making statements</li> <li>2.8 Describe how to use relational and equality operators.</li>   <li>3.1 Differentiate between applets and applications</li> <li>3.2 Illustrate some of java's exciting capabilities through the Java software development kit's demonstration applets</li> <li>3.3 Demonstrate how to write simple java applets</li> <li>3.4 Illustrate how to write a simple hypertext markup language (html) document to load an applet into the <b>appletviewer</b> or a web browser and execute the applet</li> <li>3.5 Describe the difference between variables and references.</li>   <li>4.1 Describe basic problem-solving</li> </ol>

Tel: 0044 7423211037

Email: [info@londoncomputercollege.co.uk](mailto:info@londoncomputercollege.co.uk) Website: [www.londoncomputercollege.co.uk](http://www.londoncomputercollege.co.uk)




Registered No: 3267009 (England)

<p>build and manipulate objects.</p>	<p>techniques</p> <p>4.2 Describe how to develop algorithms through the process of top-down, stepwise refinement</p> <p>4.3 Describe how to use the <b>if</b> and <b>if...else</b> selection statements to choose among alternative actions</p> <p>4.4 Illustrate how to use the <b>while</b> repetition statement to execute statements in a program repeatedly</p> <p>4.5 Describe counter-controlled repetition and sentinel-controlled repetition</p> <p>4.6 Illustrate how to use the assignment, increment and decrement operators.</p>
<p>5. Describe counter-controlled repetition. Define logical operators.</p>	<p>5.1 Describe how to use the <b>for</b> and <b>do...while</b> repetition statements to execute statements in a program repeatedly</p> <p>5.2 Illustrate multiple selection using the <b>switch</b> selection statement</p> <p>5.3 Describe how to use the <b>break</b> and <b>continue</b> program control statements</p> <p>5.4 Describe how to use the logical operators.</p>
<p>6. Define methods in Java. Describe method declarations and method overloading.</p>	<p>6.1 Describe how to construct programs modularly from small pieces called <i>methods</i></p> <p>6.2 Define the common math methods available in the java API</p> <p>6.3 Describe how to create new methods understand the mechanisms for passing information between methods</p> <p>6.4 Describe simulation techniques that use random-number generation</p> <p>6.5 Illustrate how the visibility of declarations is limited to specific regions of programs</p> <p>6.6 Describe how to write and use methods that call themselves.</p>
<p>7. Differentiate arrays and variables. Describe the process of declaring, creating and referencing arrays.</p>	<p>7.1 Define the array data structure</p> <p>7.2 Describe the use of arrays to store, sort and search lists and tables of values</p> <p>7.3 Define how to declare an array, initialise an array and refer to individual elements of an array</p> <p>7.4 Describe how to pass arrays to methods</p> <p>7.5 Illustrate how to declare and manipulate multidimensional arrays.</p>
<p>8. Define Object Oriented Programming (OOP). Describe data encapsulation and methods. Discuss how to create and use objects.</p>	<p>8.1 Describe encapsulation and data hiding;</p> <p>8.2 Describe the notions of data abstraction and abstract data types (ADTs);</p> <p>8.3 Describe how to create java ADTs—</p>

<p>9. Define inheritance. Define superclass and subclasses. Identify the relationship between them.</p> <p>10. Define polymorphism. Understand relationships among objects in an inheritance hierarchy.</p> <p>11. Define character and string classes in Java.</p> <p>12. Define Java files and streams. Illustrate how to create, read and update sequential-access files.</p>	<p>namely, classes</p> <p>8.4 Describe how to create and use objects</p> <p>8.5 Define how to control access to instance variables and methods</p> <p>8.6 Define the use of the <b>this</b> reference</p> <p>8.7 Describe how to use class variables and methods</p> <p>9.1 Describe how inheritance promotes software reusability</p> <p>9.2 Define the notions of superclasses and subclasses</p> <p>9.3 Define access modifier <b>protected</b></p> <p>9.4 Illustrate how to access superclass members with <b>super</b></p> <p>9.5 Describe the use of constructors and finalisers in inheritance hierarchies</p> <p>9.6 Demonstrates the mechanics of inheritance.</p> <p>10.1 Describe the concept of polymorphism</p> <p>10.2 Illustrate how to use overridden methods to effect polymorphism</p> <p>10.3 Distinguish between abstract and concrete classes</p> <p>10.4 Identify how to declare <b>abstract</b> methods to create abstract classes</p> <p>10.5 Define how polymorphism makes systems extensible and maintainable</p> <p>10.6 Determine an object's type at execution time.</p> <p>11.1 Demonstrate how to create and manipulate nonmodifiable character string objects of class <b>string</b></p> <p>11.2 Demonstrate how to create and manipulate modifiable character string objects of class <b>stringbuffer</b></p> <p>11.3 Illustrate how to create and manipulate objects of class <b>character</b></p> <p>11.4 Illustrate how to use a <b>stringtokenizer</b> object to break a <b>string</b> object into tokens.</p> <p>12.1 Describe how to create, read, write and update files</p> <p>12.2 Be able to use class <b>file</b></p> <p>12.3 Describe the java streams class hierarchy</p> <p>12.4 Be able to use the <b>fileinputstream</b> and <b>fileoutputstream</b> classes</p> <p>12.5 Be able to use a <b>jfilechooser</b> dialog to access files and directories</p> <p>12.6 Be able to use the</p>
--	--

	<p><b>objectinputstream</b> and <b>objectoutputstream</b> classes</p> <p>12.7 Be able to use class <b>randomaccessfile</b></p> <p>12.8 Familiarise with sequential-access and random-access file processing.</p>
--	--

### Recommended Learning Resources: Java Programming

<b>Text Books</b>	<ul style="list-style-type: none"> <li>• Effective Java: A Programming Language Guide by Joshua Bloch. ISBN-10: 0321356683</li> <li>• Java: How to Program by Harvey &amp; Paul Deitel &amp; Deitel. ISBN-10: 0132222205</li> </ul>
<b>Study Manuals</b> 	BCE produced study packs
<b>CD ROM</b> 	Power-point slides
<b>Software</b> 	Java Programming Language