






**Advanced Diploma in Programming (602) - C**

<b>Prerequisites:</b> Programming experience in C for at least six months.	<b>Corequisites:</b> A pass or higher in Diploma in Programming or equivalence.
<b>Aim:</b> The course concentrates on the design, implementation, and use of data structures in C language. Building advanced data structures based on primitive data types will be illustrated. Theoretical issues as well as examples relating to practical applications will be discussed. Emphasis will be on programming, using and improving different data structures. Understanding the complexities of different algorithms helps participants write efficient programs.	
<b>Required Materials:</b> Recommended Learning Resources.	<b>Supplementary Materials:</b> Lecture notes and tutor extra reading recommendations.
<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.	
<b>Intended Learning Outcomes:</b> 1. Define characters and strings. Identify string functions.  2. Discuss standard input and output stream. Describe conversion and field width specifier.  3. Describe how to aggregate variables under one name. Identify the different methods of defining structures.  4. Discuss data structures. Identify linked lists, stacks and queues.	<b>Assessment Criteria:</b> 1.1 Identify how to use the functions of the character handling library (ctype) 1.2 Identify how to use the string and character input/output functions of the standard input/output library (stdio) 1.3 Demonstrate how to use the string conversion functions of the general utilities library (stdlib) 1.4 Demonstrate how to use the string processing functions of the string handling library (string) 1.5 Appreciate the power of function libraries as a means of achieving software reusability.  2.1 Describe input and output streams 2.2 Demonstrate how to use all print formatting capabilities 2.3 Demonstrate how to use all input formatting capabilities  3.1 Describe how to create and use structures, unions and enumerations 3.2 Demonstrate how to pass structures to functions call by value and call by reference 3.3 Identify how to manipulate data with the bitwise operators 3.4 Describe how to create bit fields for storing data compactly.  4.1 Describe how to allocate and free memory dynamically for data objects 4.2 Demonstrate how to form linked data structures using pointers, self-referential structures and recursion 4.3 Define how to create and manipulate linked lists, queues, stacks and binary trees

5. Describe preprocess directives.	4.4	Demonstrate various important applications of linked data structures.
	5.1	Describe how to use #include for developing large programs
	5.2	Demonstrate how to use #define to create macros and macros with arguments
	5.3	Define conditional compilation
	5.4	Describe how to display error messages during conditional compilation
	5.5	Demonstrate to use assertions to test if the values of expressions are correct

### **Recommended Learning Resources: Advanced C Programming**

<b>Text Books</b>	<ul style="list-style-type: none"> <li>The C Programming Language by Brian W. Kernighan and Dennis Ritchie. ISBN-10: 0131103628</li> <li>Advanced C Programming by Example by John Perry. ISBN-10: 0534951406</li> <li>Advanced C. Programming by Waite Group. ISBN-10: 0893034738</li> </ul>
<b>Study Manuals</b> 	BCE produced study packs
<b>CD ROM</b> 	Power-point slides
<b>Software</b> 	C Programming Language