



**Yashwantrao Chavan  
Maharashtra Open  
University**

**AEC111**

**English Communication**

### **Brief Contents**

|   |            |
|---|------------|
| <b>Vice Chancellor's Message.....</b>                             | <b>3</b>   |
| <b>Forward By The Director .....</b>                              | <b>4</b>   |
| <b>Credit 01 .....</b>  | <b>5</b>   |
| <b>Unit 01-01: Introduction.....</b>                              | <b>6</b>   |
| <b>Unit 01-02: Verbal and Non-Verbal Communication .....</b>      | <b>19</b>  |
| <b>Unit 01-03: Working with Customers .....</b>                   | <b>43</b>  |
| <b>Unit 01-04: Developing Professional Telephone Skills .....</b> | <b>56</b>  |
| <b>Unit 01-05: Improving Informal Communication .....</b>         | <b>66</b>  |
| <b>Credit 02.....</b>   | <b>90</b>  |
| <b>Unit 02-01: Understanding Reading and Writing .....</b>        | <b>91</b>  |
| <b>Unit 02-02: Uncovering the Secrets of Clear Writing .....</b>  | <b>99</b>  |
| <b>Unit 02-03: Communicating with E-Mail and Memos .....</b>      | <b>115</b> |
| <b>Unit 02-04: Developing Reports and Proposals .....</b>         | <b>126</b> |
| <b>Unit 02-05: Writing for Employment.....</b>                    | <b>136</b> |
| <b>Credit 03.....</b>   | <b>151</b> |
| <b>Unit 03-01: Identifying and Defining Problems .....</b>        | <b>152</b> |
| <b>Unit 03-02: Solving the Problem .....</b>                      | <b>170</b> |
| <b>Unit 03-03: Thinking Critically .....</b>                      | <b>184</b> |
| <b>Unit 03-04: Group Decision Making and Problem Solving.....</b> | <b>195</b> |
| <b>Credit 04.....</b>   | <b>206</b> |
| <b>Unit 04-01: Working in Groups and Teams .....</b>              | <b>207</b> |
| <b>Unit 04-02: Exploring Team Roles and Processes.....</b>        | <b>217</b> |
| <b>Unit 04-03: Building and Developing Teams .....</b>            | <b>234</b> |
| <b>Unit 04-04: Presenting Yourself Professionally .....</b>       | <b>245</b> |
| <b>Unit 04-05: Developing Your Interpersonal Skills.....</b>      | <b>254</b> |
| <b>Answers to Self-Tests.....</b>                                 | <b>264</b> |

## Contents

### **Lecture 1: C Programming Introduction**

- Why C
- Constants, Variables and Keywords
- Rules for building variables and constants
- Keywords in C
- In-memory view of a variable

### **Lecture 2: The First C Program**

- Adding comments to a program
- Creating a simple C program
- The general form of `printf()`

### **Lecture 3: C Instructions Part I**

- Editors, Compilers, Linkers, Preprocessor, etc.
- Using Integrated Development Environment
- Editing commands

### **Lecture 4: C Instructions Part II**

- Type declaration instructions
- Arithmetic instructions
- Associativity
- Numbering systems
- Escape sequences

### **Lecture 5: Control Instructions Part I**

- Control instructions
- Decision control instructions

### **Lecture 6: Control Instructions Part II**

- The `sizeof()` operator
- Relational operators
- The `if-else` statement

### **Lecture 7: More Decision Control Instructions Part I**

- Logical operators
- Usage of logical operators
- Hierarchy of operators

### **Lecture 8: More Decision Control Instructions Part II**

- Conditional operators

**Lecture 9: Conversions**

- Conversions from one numbering system to another
- ASCII codes
- Printing graphics characters

**Lecture 10: While loop I**

- Repetition control instructions
- Incrementation & Decrementation operators

**Lecture 11: While loop II**

- Pre & Post incrementation/decrementation
- Loop control instructions
- The **while** loop

**Lecture 12: For loop**

- The **for** loop
- Differences between **while** and **for** loops

**Lecture 13: DoWhile loop**

- Multiple initializations in the **for** statement
- Multiple incrementations in the **for** statement
- Implement the **do – while** loop
- Difference between **for**, **while** and **do-while** loops
- Use of **break** and **continue** statements

**Lecture 14: Switch**

- How to use switch-case control instruction

**Lecture 15: Menu**

- Menu driven programming
- Function definition
- Function declaration
- Function calls

**Lecture 16: Functions**

- Passing values to the functions
- Returning a value from the function

**Lecture 17: Advanced Features**

- Returning non-integer values from the functions
- Pointers
- Printing address of variables

**Lecture 18: Pointers**

- Address of operator ( & )
- Value at the address operator ( \* )
- Accessing VDU memory using pointers

**Lecture 19: More Pointers I**

- Accesses VDU memory directly
- Playing sound

**Lecture 20: More Pointers II**

- Toggle keys
- Switching On & Off Caps lock programmatically
- Restart computer without physically depressing Ctrl + Alt keys

**Lecture 21: Near Far**

- Difference between *far* and *near* pointer.
- Size of *far* and *near* pointer.
- Segment:Offset addressing scheme

**Lecture 22: Far Huge**

- Huge pointers
- Finding the size of base memory

**Lecture 23: Call By Value & Reference**

- The difference between call by value and call by reference
- How to return more than one value from a function

**Lecture 24: Recursion**

- Recursion
- Find Factorial Value Using Recursion

**Lecture 25: Data Types**

- Data types, size and range
- The type of **ints**, **float** and **char** data types

**Lecture 26: Storage Classes**

- The storage classes
- Automatic storage class
- Register storage class
- Static storage class
- External storage class

**Lecture 27: Preprocessor I**

- Compiling and linking of a program

*iii*

- A preprocessor
- Types of preprocessor directives
- Preprocessor directives and their use
- Macros
- Difference between macro templates and macro expansion

**Lecture 28: PreProcessor II**

- Macros with arguments
- Condition compilation
- Miscellaneous preprocessor directives

**Lecture 29: Arrays**

- Arrays
- Declare and initialize array

**Lecture 30: Sorting**

- Sorting techniques

**Lecture 31: 2D Arrays I**

- Accessing array using pointers
- Pointers and arithmetic operations
- Passing array-elements to a function
- Passing entire array to a function

**Lecture 32: 2D Arrays II**

- In-memory organization of an array
- 2-Dimensional arrays
- Pointers and multidimensional array

**Lecture 33: Application of Arrays I**

- Create 2-D arrays
- Representing elements of 1-D and multidimensional array using pointer

**Lecture 34: Application of Arrays II**

- Obtaining scan codes of keys
- Creating a 3-D array
- A puzzle game

**Lecture 35: Strings I**

- Strings
- Creating strings
- Handling multiword string
- Difference between **printf()** and **puts()**

**Lecture 36: Strings II**

- Finding length of a string
- Copying one string to another
- Concatenation of strings
- Converting string from lower to uppercase
- Usage of standard library string functions

**Lecture 37: Strings III**

- Const pointers
- Need of **const** pointers
- 2-D array of characters
- Creating array of pointers to strings

**Lecture 38: Calender**

- Creating a monthly calendar

**Lecture 39: Structures I**

- Structures
- Access elements of a structure
- Array of structures

**Lecture 40: Structures II**

- Nested structures
- Passing elements of a structure to a function
- Passing structure to a function
- Returning structure from a function

**Lecture 41: Data Structures I**

- Disadvantages of arrays
- Disadvantages of static memory allocation
- Allocating memory dynamically
- Implementing a linked list

**Lecture 42: Data Structures II**

- Stack
- Implementing stack as linked list

**Lecture 43: Disk I**

- Organization of a Floppy disk and Hard disk
- Reading contents of boot sector
- Boot Parameters

**Lecture 44: Disk II**

- Read a Disk
- Viral Infected Disk
- Anti Viral Solution

**Lecture 45: Directory**

- The directory structure
- Reading directory sectors
- Date and time of a file in directory structure
- File Attributes
- Loading and saving of a file

**Lecture 46: Console I/O**

- Console I/O functions
- Difference between formatted and unformatted Console I/O

**Lecture 47: File I/O I**

- Disk I/O operations
- Buffered I/O
- Read a file and displaying its contents
- Using `typedef`

**Lecture 48: File I/O II**

- Writing data to a file
- Copying contents of one file to another
- Encoding and Decoding of file contents

**Lecture 49: More File I/O I**

- Program remove blank lines from a file
- To read and write records from/to the file
- Various file opening modes
- Difference between text mode and binary mode file I/O

**Lecture 50: More File I/O II**

- To recover data from a virus infected file
- To change an internal DOS command called DIR
- To perform low-level Disk I/O

**Lecture 51: Miscellany**

- Introduction to Enumerations, Structures and Unions
- Bitwise operators
- The utility of << (Left-Shift) and >> (Right-Shift) operators

**Lecture 52: Bitwise Operators**

- The utility of & (Bitwise AND) operator
- To change attributes of a file using bitwise operators
- To create function pointers

**Lecture 53: Hardware Interaction I**

- Ways to interact with hardware
- Hardware interaction, DOS perspective
- Hardware interaction, Windows perspective
- Ports, CMOS
- Accessing CMOS data

**Lecture 54: Hardware Interaction II**

- Serial communication using null modem
- Communication with parallel port programmatically
- Working of Speaker
- Playing tunes using ports

**Lecture 55: Windows I**

- Advantages of Windows Programming model over DOS model

**Lecture 56: Windows II**

- Creating a window

**Lecture 57: Windows III**

- Event Driven Programming Model
- To create and display a real world window

**Lecture 58: Windows IV**

- Device Context
- Graphics under Window

**Lecture 59: Windows V**

- Freehand drawing using mouse
- To Capture Mouse
- To displaying Bitmap
- Animation

**Lecture 60: Linux**

- Introduction to Linux
- C programming under Linux
- Parent and Child Processes
- Communication using Signals

## Contents

|   |           |
|---|-----------|
| <b>Lecture 1: C Programming Introduction</b>  | <b>01</b> |
| <ul style="list-style-type: none"> <li>• Why C</li> <li>• Constants, Variables and Keywords</li> <li>• Rules for building variables and constants</li> <li>• Keywords in C</li> <li>• In-memory view of a variable</li> </ul> |           |
| <b>Lecture 2: The First C Program</b>   | <b>05</b> |
| <ul style="list-style-type: none"> <li>• Adding comments to a program</li> <li>• Creating a simple C program</li> <li>• The general form of <code>printf()</code></li> </ul>  |           |
| <b>Lecture 3: C Instructions Part I</b>   | <b>09</b> |
| <ul style="list-style-type: none"> <li>• Editors, Compilers, Linkers, Preprocessor, etc.</li> <li>• Using Integrated Development Environment</li> <li>• Editing commands</li> </ul>   |           |
| <b>Lecture 4: C Instructions Part II</b>  | <b>13</b> |
| <ul style="list-style-type: none"> <li>• Type declaration instructions</li> <li>• Arithmetic instructions</li> <li>• Associativity</li> <li>• Numbering systems</li> <li>• Escape sequences</li> </ul>                        |           |
| <b>Lecture 5: Control Instructions Part I</b>   | <b>19</b> |
| <ul style="list-style-type: none"> <li>• Control instructions</li> <li>• Decision control instructions</li> </ul>   |           |
| <b>Lecture 6: Control Instructions Part II</b>  | <b>23</b> |
| <ul style="list-style-type: none"> <li>• The <code>sizeof()</code> operator</li> <li>• Relational operators</li> <li>• The <code>if-else</code> statement</li> </ul>  |           |
| <b>Lecture 7: More Decision Control Instructions Part I</b>   | <b>27</b> |
| <ul style="list-style-type: none"> <li>• Logical operators</li> <li>• Usage of logical operators</li> <li>• Hierarchy of operators</li> </ul>   |           |
| <b>Lecture 8: More Decision Control Instructions Part II</b>  | <b>31</b> |
| <ul style="list-style-type: none"> <li>• Conditional operators</li> </ul>   |           |

|   |           |
|---|-----------|
| <b>Lecture 9: Conversions</b>   | <b>35</b> |
| <ul style="list-style-type: none"> <li>• Conversions from one numbering system to another</li> <li>• ASCII codes</li> <li>• Printing graphics characters</li> </ul>   |           |
| <b>Lecture 10: While loop I</b>   | <b>39</b> |
| <ul style="list-style-type: none"> <li>• Repetition control instructions</li> <li>• Incrementation &amp; Decrementation operators</li> </ul>  |           |
| <b>Lecture 11: While loop II</b>  | <b>43</b> |
| <ul style="list-style-type: none"> <li>• Pre &amp; Post incrementation/decrementation</li> <li>• Loop control instructions</li> <li>• The <b>while</b> loop</li> </ul>  |           |
| <b>Lecture 12: For loop</b>   | <b>47</b> |
| <ul style="list-style-type: none"> <li>• The <b>for</b> loop</li> <li>• Differences between <b>while</b> and <b>for</b> loops</li> </ul>  |           |
| <b>Lecture 13: DoWhile loop</b>   | <b>51</b> |
| <ul style="list-style-type: none"> <li>• Multiple initializations in the <b>for</b> statement</li> <li>• Multiple incrementations in the <b>for</b> statement</li> <li>• Implement the <b>do – while</b> loop</li> <li>• Difference between <b>for</b>, <b>while</b> and <b>do-while</b> loops</li> <li>• Use of <b>break</b> and <b>continue</b> statements</li> </ul> |           |
| <b>Lecture 14: Switch</b>   | <b>55</b> |
| <ul style="list-style-type: none"> <li>• How to use switch-case control instruction</li> </ul>  |           |
| <b>Lecture 15: Menu</b>   | <b>59</b> |
| <ul style="list-style-type: none"> <li>• Menu driven programming</li> <li>• Function definition</li> <li>• Function declaration</li> <li>• Function calls</li> </ul>  |           |
| <b>Lecture 16: Functions</b>  | <b>63</b> |
| <ul style="list-style-type: none"> <li>• Passing values to the functions</li> <li>• Returning a value from the function</li> </ul>  |           |
| <b>Lecture 17: Advanced Features</b>  | <b>67</b> |
| <ul style="list-style-type: none"> <li>• Returning non-integer values from the functions</li> <li>• Pointers</li> <li>• Printing address of variables</li> </ul>  |           |

|  |            |
|--|------------|
| <b>Lecture 18: Pointers</b>  | <b>71</b>  |
| <ul style="list-style-type: none"> <li>• Address of operator ( &amp; )</li> <li>• Value at the address operator ( * )</li> <li>• Accessing VDU memory using pointers</li> </ul>                                |            |
| <b>Lecture 19: More Pointers I</b>   | <b>75</b>  |
| <ul style="list-style-type: none"> <li>• Accesses VDU memory directly</li> <li>• Playing sound</li> </ul>  |            |
| <b>Lecture 20: More Pointers II</b>  | <b>77</b>  |
| <ul style="list-style-type: none"> <li>• Toggle keys</li> <li>• Switching On &amp; Off Caps lock programmatically</li> <li>• Restart computer without physically depressing Ctrl + Alt keys</li> </ul>         |            |
| <b>Lecture 21: Near Far</b>  | <b>79</b>  |
| <ul style="list-style-type: none"> <li>• Difference between <i>far</i> and <i>near</i> pointer.</li> <li>• Size of <i>far</i> and <i>near</i> pointer.</li> <li>• Segment:Offset addressing scheme</li> </ul>  |            |
| <b>Lecture 22: Far Huge</b>  | <b>83</b>  |
| <ul style="list-style-type: none"> <li>• Huge pointers</li> <li>• Finding the size of base memory</li> </ul>   |            |
| <b>Lecture 23: Call By Value &amp; Reference</b>   | <b>85</b>  |
| <ul style="list-style-type: none"> <li>• The difference between call by value and call by reference</li> <li>• How to return more than one value from a function</li> </ul>                                    |            |
| <b>Lecture 24: Recursion</b>   | <b>89</b>  |
| <ul style="list-style-type: none"> <li>• Recursion</li> <li>• Find Factorial Value Using Recursion</li> </ul>  |            |
| <b>Lecture 25: Data Types</b>  | <b>93</b>  |
| <ul style="list-style-type: none"> <li>• Data types, size and range</li> <li>• The type of <b>ints</b>, <b>float</b> and <b>char</b> data types</li> </ul>   |            |
| <b>Lecture 26: Storage Classes</b>   | <b>97</b>  |
| <ul style="list-style-type: none"> <li>• The storage classes</li> <li>• Automatic storage class</li> <li>• Register storage class</li> <li>• Static storage class</li> <li>• External storage class</li> </ul> |            |
| <b>Lecture 27: Preprocessor I</b>  | <b>101</b> |
| <ul style="list-style-type: none"> <li>• Compiling and linking of a program</li> </ul>   |            |

|   |            |
|---|------------|
| • A preprocessor  |            |
| • Types of preprocessor directives                                      |            |
| • Preprocessor directives and their use                                 |            |
| • Macros  |            |
| • Difference between macro templates and macro expansion                |            |
| <b>Lecture 28: PreProcessor II</b>                                      | <b>105</b> |
| • Macros with arguments   |            |
| • Condition compilation   |            |
| • Miscellaneous preprocessor directives                                 |            |
| <b>Lecture 29: Arrays</b>   | <b>109</b> |
| • Arrays  |            |
| • Declare and initialize array  |            |
| <b>Lecture 30: Sorting</b>  | <b>113</b> |
| • Sorting techniques  |            |
| <b>Lecture 31: 2D Arrays I</b>  | <b>115</b> |
| • Accessing array using pointers  |            |
| • Pointers and arithmetic operations                                    |            |
| • Passing array-elements to a function                                  |            |
| • Passing entire array to a function                                    |            |
| <b>Lecture 32: 2D Arrays II</b>   | <b>119</b> |
| • In-memory organization of an array                                    |            |
| • 2-Dimensional arrays  |            |
| • Pointers and multidimensional array                                   |            |
| <b>Lecture 33: Application of Arrays I</b>                              | <b>123</b> |
| • Create 2-D arrays   |            |
| • Representing elements of 1-D and multidimensional array using pointer |            |
| <b>Lecture 34: Application of Arrays II</b>                             | <b>125</b> |
| • Obtaining scan codes of keys  |            |
| • Creating a 3-D array  |            |
| • A puzzle game   |            |
| <b>Lecture 35: Strings I</b>  | <b>129</b> |
| • Strings   |            |
| • Creating strings  |            |
| • Handling multiword string   |            |
| • Difference between <b>printf()</b> and <b>puts()</b>                  |            |

|  |            |
|--|------------|
| <b>Lecture 36: Strings II</b>  | <b>133</b> |
| <ul style="list-style-type: none"> <li>• Finding length of a string</li> <li>• Copying one string to another</li> <li>• Concatenation of strings</li> <li>• Converting string from lower to uppercase</li> <li>• Usage of standard library string functions</li> </ul> |            |
| <b>Lecture 37: Strings III</b>   | <b>137</b> |
| <ul style="list-style-type: none"> <li>• Const pointers</li> <li>• Need of <b>const</b> pointers</li> <li>• 2-D array of characters</li> <li>• Creating array of pointers to strings</li> </ul>  |            |
| <b>Lecture 38: Calender</b>  | <b>141</b> |
| <ul style="list-style-type: none"> <li>• Creating a monthly calendar</li> </ul>  |            |
| <b>Lecture 39: Structures I</b>  | <b>145</b> |
| <ul style="list-style-type: none"> <li>• Structures</li> <li>• Access elements of a structure</li> <li>• Array of structures</li> </ul>  |            |
| <b>Lecture 40: Structures II</b>   | <b>149</b> |
| <ul style="list-style-type: none"> <li>• Nested structures</li> <li>• Passing elements of a structure to a function</li> <li>• Passing structure to a function</li> <li>• Returning structure from a function</li> </ul>   |            |
| <b>Lecture 41: Data Structures I</b>   | <b>153</b> |
| <ul style="list-style-type: none"> <li>• Disadvantages of arrays</li> <li>• Disadvantages of static memory allocation</li> <li>• Allocating memory dynamically</li> <li>• Implementing a linked list</li> </ul>  |            |
| <b>Lecture 42: Data Structures II</b>  | <b>157</b> |
| <ul style="list-style-type: none"> <li>• Stack</li> <li>• Implementing stack as linked list</li> </ul>   |            |
| <b>Lecture 43: Disk I</b>  | <b>161</b> |
| <ul style="list-style-type: none"> <li>• Organization of a Floppy disk and Hard disk</li> <li>• Reading contents of boot sector</li> <li>• Boot Parameters</li> </ul>  |            |

|   |            |
|---|------------|
| <b>Lecture 44: Disk II</b>                                      | <b>165</b> |
| • Read a Disk   |            |
| • Viral Infected Disk   |            |
| • Anti Viral Solution   |            |
| <b>Lecture 45: Directory</b>                                    | <b>167</b> |
| • The directory structure                                       |            |
| • Reading directory sectors                                     |            |
| • Date and time of a file in directory structure                |            |
| • File Attributes   |            |
| • Loading and saving of a file                                  |            |
| <b>Lecture 46: Console I/O</b>                                  | <b>171</b> |
| • Console I/O functions   |            |
| • Difference between formatted and unformatted Console I/O      |            |
| <b>Lecture 47: File I/O I</b>                                   | <b>175</b> |
| • Disk I/O operations   |            |
| • Buffered I/O  |            |
| • Read a file and displaying its contents                       |            |
| • Using <b>typedef</b>  |            |
| <b>Lecture 48: File I/O II</b>                                  | <b>179</b> |
| • Writing data to a file  |            |
| • Copying contents of one file to another                       |            |
| • Encoding and Decoding of file contents                        |            |
| <b>Lecture 49: More File I/O I</b>                              | <b>183</b> |
| • Program remove blank lines from a file                        |            |
| • To read and write records from/to the file                    |            |
| • Various file opening modes                                    |            |
| • Difference between text mode and binary mode file I/O         |            |
| <b>Lecture 50: More File I/O II</b>                             | <b>187</b> |
| • To recover data from a virus infected file                    |            |
| • To change an internal DOS command called DIR                  |            |
| • To perform low-level Disk I/O                                 |            |
| <b>Lecture 51: Miscellany</b>                                   | <b>191</b> |
| • Introduction to Enumerations, Structures and Unions           |            |
| • Bitwise operators   |            |
| • The utility of << (Left-Shift) and >> (Right-Shift) operators |            |

|  |            |
|--|------------|
| <b>Lecture 52: Bitwise Operators</b>   | <b>195</b> |
| <ul style="list-style-type: none"> <li>• The utility of &amp; (Bitwise AND) operator</li> <li>• To change attributes of a file using bitwise operators</li> <li>• To create function pointers</li> </ul>                                       |            |
| <b>Lecture 53: Hardware Interaction I</b>  | <b>199</b> |
| <ul style="list-style-type: none"> <li>• Ways to interact with hardware</li> <li>• Hardware interaction, DOS perspective</li> <li>• Hardware interaction, Windows perspective</li> <li>• Ports, CMOS</li> <li>• Accessing CMOS data</li> </ul> |            |
| <b>Lecture 54: Hardware Interaction II</b>   | <b>205</b> |
| <ul style="list-style-type: none"> <li>• Serial communication using null modem</li> <li>• Communication with parallel port programmatically</li> <li>• Working of Speaker</li> <li>• Playing tunes using ports</li> </ul>                      |            |
| <b>Lecture 55: Windows I</b>   | <b>211</b> |
| <ul style="list-style-type: none"> <li>• Advantages of Windows Programming model over DOS model</li> </ul>   |            |
| <b>Lecture 56: Windows II</b>  | <b>215</b> |
| <ul style="list-style-type: none"> <li>• Creating a window</li> </ul>  |            |
| <b>Lecture 57: Windows III</b>   | <b>217</b> |
| <ul style="list-style-type: none"> <li>• Event Driven Programming Model</li> <li>• To create and display a real world window</li> </ul>  |            |
| <b>Lecture 58: Windows IV</b>  | <b>223</b> |
| <ul style="list-style-type: none"> <li>• Device Context</li> <li>• Graphics under Window</li> </ul>  |            |
| <b>Lecture 59: Windows V</b>   | <b>227</b> |
| <ul style="list-style-type: none"> <li>• Freehand drawing using mouse</li> <li>• To Capture Mouse</li> <li>• To displaying Bitmap</li> <li>• Animation</li> </ul>  |            |
| <b>Lecture 60: Linux</b>   | <b>233</b> |
| <ul style="list-style-type: none"> <li>• Introduction to Linux</li> <li>• C programming under Linux</li> <li>• Parent and Child Processes</li> <li>• Communication using Signals</li> </ul>  |            |

## Contents

|   |           |
|---|-----------|
| <b>Lecture 1: C Programming Introduction</b>  | <b>01</b> |
| <ul style="list-style-type: none"> <li>• Why C</li> <li>• Constants, Variables and Keywords</li> <li>• Rules for building variables and constants</li> <li>• Keywords in C</li> <li>• In-memory view of a variable</li> </ul> |           |
| <b>Lecture 2: The First C Program</b>   | <b>03</b> |
| <ul style="list-style-type: none"> <li>• Adding comments to a program</li> <li>• Creating a simple C program</li> <li>• The general form of <b>printf()</b></li> </ul>  |           |
| <b>Lecture 3: C Instructions Part I</b>   | <b>05</b> |
| <ul style="list-style-type: none"> <li>• Editors, Compilers, Linkers, Preprocessor, etc.</li> <li>• Using Integrated Development Environment</li> <li>• Editing commands</li> </ul>   |           |
| <b>Lecture 4: C Instructions Part II</b>  | <b>07</b> |
| <ul style="list-style-type: none"> <li>• Type declaration instructions</li> <li>• Arithmetic instructions</li> <li>• Associativity</li> <li>• Numbering systems</li> <li>• Escape sequences</li> </ul>                        |           |
| <b>Lecture 5: Control Instructions Part I</b>   | <b>09</b> |
| <ul style="list-style-type: none"> <li>• Control instructions</li> <li>• Decision control instructions</li> </ul>   |           |
| <b>Lecture 6: Control Instructions Part II</b>  | <b>11</b> |
| <ul style="list-style-type: none"> <li>• The <b>sizeof()</b> operator</li> <li>• Relational operators</li> <li>• The <b>if-else</b> statement</li> </ul>  |           |
| <b>Lecture 7: More Decision Control Instructions Part I</b>   | <b>13</b> |
| <ul style="list-style-type: none"> <li>• Logical operators</li> <li>• Usage of logical operators</li> <li>• Hierarchy of operators</li> </ul>   |           |
| <b>Lecture 8: More Decision Control Instructions Part II</b>  | <b>17</b> |
| <ul style="list-style-type: none"> <li>• Conditional operators</li> </ul>   |           |

|   |           |
|---|-----------|
| <b>Lecture 9: Conversions</b>   | <b>19</b> |
| <ul style="list-style-type: none"> <li>• Conversions from one numbering system to another</li> <li>• ASCII codes</li> <li>• Printing graphics characters</li> </ul>   |           |
| <b>Lecture 10: While loop I</b>   | <b>21</b> |
| <ul style="list-style-type: none"> <li>• Repetition control instructions</li> <li>• Incrementation &amp; Decrementation operators</li> </ul>  |           |
| <b>Lecture 11: While loop II</b>  | <b>25</b> |
| <ul style="list-style-type: none"> <li>• Pre &amp; Post incrementation/decrementation</li> <li>• Loop control instructions</li> <li>• The <b>while</b> loop</li> </ul>  |           |
| <b>Lecture 12: For loop</b>   | <b>27</b> |
| <ul style="list-style-type: none"> <li>• The <b>for</b> loop</li> <li>• Differences between <b>while</b> and <b>for</b> loops</li> </ul>  |           |
| <b>Lecture 13: DoWhile loop</b>   | <b>29</b> |
| <ul style="list-style-type: none"> <li>• Multiple initializations in the <b>for</b> statement</li> <li>• Multiple incrementations in the <b>for</b> statement</li> <li>• Implement the <b>do – while</b> loop</li> <li>• Difference between <b>for</b>, <b>while</b> and <b>do-while</b> loops</li> <li>• Use of <b>break</b> and <b>continue</b> statements</li> </ul> |           |
| <b>Lecture 14: Switch</b>   | <b>31</b> |
| <ul style="list-style-type: none"> <li>• How to use switch-case control instruction</li> </ul>  |           |
| <b>Lecture 15: Menu</b>   | <b>35</b> |
| <ul style="list-style-type: none"> <li>• Menu driven programming</li> <li>• Function definition</li> <li>• Function declaration</li> <li>• Function calls</li> </ul>  |           |
| <b>Lecture 16: Functions</b>  | <b>37</b> |
| <ul style="list-style-type: none"> <li>• Passing values to the functions</li> <li>• Returning a value from the function</li> </ul>  |           |
| <b>Lecture 17: Advanced Features</b>  | <b>41</b> |
| <ul style="list-style-type: none"> <li>• Returning non-integer values from the functions</li> <li>• Pointers</li> <li>• Printing address of variables</li> </ul>  |           |

|  |           |
|--|-----------|
| <b>Lecture 18: Pointers</b>  | <b>43</b> |
| <ul style="list-style-type: none"> <li>• Address of operator ( &amp; )</li> <li>• Value at the address operator ( * )</li> <li>• Accessing VDU memory using pointers</li> </ul>                                |           |
| <b>Lecture 19: More Pointers I</b>   | <b>45</b> |
| <ul style="list-style-type: none"> <li>• Accesses VDU memory directly</li> <li>• Playing sound</li> </ul>  |           |
| <b>Lecture 20: More Pointers II</b>  | <b>47</b> |
| <ul style="list-style-type: none"> <li>• Toggle keys</li> <li>• Switching On &amp; Off Caps lock programmatically</li> <li>• Restart computer without physically depressing Ctrl + Alt keys</li> </ul>         |           |
| <b>Lecture 21: Near Far</b>  | <b>49</b> |
| <ul style="list-style-type: none"> <li>• Difference between <i>far</i> and <i>near</i> pointer.</li> <li>• Size of <i>far</i> and <i>near</i> pointer.</li> <li>• Segment:Offset addressing scheme</li> </ul>  |           |
| <b>Lecture 22: Far Huge</b>  | <b>51</b> |
| <ul style="list-style-type: none"> <li>• Huge pointers</li> <li>• Finding the size of base memory</li> </ul>   |           |
| <b>Lecture 23: Call By Value &amp; Reference</b>   | <b>53</b> |
| <ul style="list-style-type: none"> <li>• The difference between call by value and call by reference</li> <li>• How to return more than one value from a function</li> </ul>                                    |           |
| <b>Lecture 24: Recursion</b>   | <b>57</b> |
| <ul style="list-style-type: none"> <li>• Recursion</li> <li>• Find Factorial Value Using Recursion</li> </ul>  |           |
| <b>Lecture 25: Data Types</b>  | <b>59</b> |
| <ul style="list-style-type: none"> <li>• Data types, size and range</li> <li>• The type of <b>ints</b>, <b>float</b> and <b>char</b> data types</li> </ul>   |           |
| <b>Lecture 26: Storage Classes</b>   | <b>61</b> |
| <ul style="list-style-type: none"> <li>• The storage classes</li> <li>• Automatic storage class</li> <li>• Register storage class</li> <li>• Static storage class</li> <li>• External storage class</li> </ul> |           |
| <b>Lecture 27: Preprocessor I</b>  | <b>63</b> |
| <ul style="list-style-type: none"> <li>• Compiling and linking of a program</li> </ul>   |           |

|   |           |
|---|-----------|
| <ul style="list-style-type: none"> <li>• A preprocessor</li> <li>• Types of preprocessor directives</li> <li>• Preprocessor directives and their use</li> <li>• Macros</li> <li>• Difference between macro templates and macro expansion</li> </ul> |           |
| <b>Lecture 28: PreProcessor II</b>  | <b>65</b> |
| <ul style="list-style-type: none"> <li>• Macros with arguments</li> <li>• Condition compilation</li> <li>• Miscellaneous preprocessor directives</li> </ul>   |           |
| <b>Lecture 29: Arrays</b>   | <b>67</b> |
| <ul style="list-style-type: none"> <li>• Arrays</li> <li>• Declare and initialize array</li> </ul>  |           |
| <b>Lecture 30: Sorting</b>  | <b>69</b> |
| <ul style="list-style-type: none"> <li>• Sorting techniques</li> </ul>  |           |
| <b>Lecture 31: 2D Arrays I</b>  | <b>71</b> |
| <ul style="list-style-type: none"> <li>• Accessing array using pointers</li> <li>• Pointers and arithmetic operations</li> <li>• Passing array-elements to a function</li> <li>• Passing entire array to a function</li> </ul>                      |           |
| <b>Lecture 32: 2D Arrays II</b>   | <b>75</b> |
| <ul style="list-style-type: none"> <li>• In-memory organization of an array</li> <li>• 2-Dimensional arrays</li> <li>• Pointers and multidimensional array</li> </ul>   |           |
| <b>Lecture 33: Application of Arrays I</b>  | <b>79</b> |
| <ul style="list-style-type: none"> <li>• Create 2-D arrays</li> <li>• Representing elements of 1-D and multidimensional array using pointer</li> </ul>  |           |
| <b>Lecture 34: Application of Arrays II</b>   | <b>81</b> |
| <ul style="list-style-type: none"> <li>• Obtaining scan codes of keys</li> <li>• Creating a 3-D array</li> <li>• A puzzle game</li> </ul>   |           |
| <b>Lecture 35: Strings I</b>  | <b>83</b> |
| <ul style="list-style-type: none"> <li>• Strings</li> <li>• Creating strings</li> <li>• Handling multiword string</li> <li>• Difference between <b>printf()</b> and <b>puts()</b></li> </ul>  |           |

|  |            |
|--|------------|
| <b>Lecture 36: Strings II</b>  | <b>85</b>  |
| <ul style="list-style-type: none"><li>• Finding length of a string</li><li>• Copying one string to another</li><li>• Concatenation of strings</li><li>• Converting string from lower to uppercase</li><li>• Usage of standard library string functions</li></ul> |            |
| <b>Lecture 37: Strings III</b>   | <b>87</b>  |
| <ul style="list-style-type: none"><li>• Const pointers</li><li>• Need of <b>const</b> pointers</li><li>• 2-D array of characters</li><li>• Creating array of pointers to strings</li></ul>   |            |
| <b>Lecture 38: Calender</b>  | <b>89</b>  |
| <ul style="list-style-type: none"><li>• Creating a monthly calendar</li></ul>  |            |
| <b>Lecture 39: Structures I</b>  | <b>91</b>  |
| <ul style="list-style-type: none"><li>• Structures</li><li>• Access elements of a structure</li><li>• Array of structures</li></ul>  |            |
| <b>Lecture 40: Structures II</b>   | <b>95</b>  |
| <ul style="list-style-type: none"><li>• Nested structures</li><li>• Passing elements of a structure to a function</li><li>• Passing structure to a function</li><li>• Returning structure from a function</li></ul>  |            |
| <b>Lecture 41: Data Structures I</b>   | <b>97</b>  |
| <ul style="list-style-type: none"><li>• Disadvantages of arrays</li><li>• Disadvantages of static memory allocation</li><li>• Allocating memory dynamically</li><li>• Implementing a linked list</li></ul>   |            |
| <b>Lecture 42: Data Structures II</b>  | <b>99</b>  |
| <ul style="list-style-type: none"><li>• Stack</li><li>• Implementing stack as linked list</li></ul>  |            |
| <b>Lecture 43: Disk I</b>  | <b>101</b> |
| <ul style="list-style-type: none"><li>• Organization of a Floppy disk and Hard disk</li><li>• Reading contents of boot sector</li><li>• Boot Parameters</li></ul>  |            |

|   |            |
|---|------------|
| <b>Lecture 44: Disk II</b>  | <b>103</b> |
| <ul style="list-style-type: none"> <li>• Read a Disk</li> <li>• Viral Infected Disk</li> <li>• Anti Viral Solution</li> </ul>   |            |
| <b>Lecture 45: Directory</b>  | <b>105</b> |
| <ul style="list-style-type: none"> <li>• The directory structure</li> <li>• Reading directory sectors</li> <li>• Date and time of a file in directory structure</li> <li>• File Attributes</li> <li>• Loading and saving of a file</li> </ul>           |            |
| <b>Lecture 46: Console I/O</b>  | <b>107</b> |
| <ul style="list-style-type: none"> <li>• Console I/O functions</li> <li>• Difference between formatted and unformatted Console I/O</li> </ul>   |            |
| <b>Lecture 47: File I/O I</b>   | <b>111</b> |
| <ul style="list-style-type: none"> <li>• Disk I/O operations</li> <li>• Buffered I/O</li> <li>• Read a file and displaying its contents</li> <li>• Using <b>typedef</b></li> </ul>  |            |
| <b>Lecture 48: File I/O II</b>  | <b>113</b> |
| <ul style="list-style-type: none"> <li>• Writing data to a file</li> <li>• Copying contents of one file to another</li> <li>• Encoding and Decoding of file contents</li> </ul>   |            |
| <b>Lecture 49: More File I/O I</b>  | <b>115</b> |
| <ul style="list-style-type: none"> <li>• Program remove blank lines from a file</li> <li>• To read and write records from/to the file</li> <li>• Various file opening modes</li> <li>• Difference between text mode and binary mode file I/O</li> </ul> |            |
| <b>Lecture 50: More File I/O II</b>   | <b>117</b> |
| <ul style="list-style-type: none"> <li>• To recover data from a virus infected file</li> <li>• To change an internal DOS command called DIR</li> <li>• To perform low-level Disk I/O</li> </ul>   |            |
| <b>Lecture 51: Miscellany</b>   | <b>119</b> |
| <ul style="list-style-type: none"> <li>• Introduction to Enumerations, Structures and Unions</li> <li>• Bitwise operators</li> <li>• The utility of &lt;&lt; (Left-Shift) and &gt;&gt; (Right-Shift) operators</li> </ul>                               |            |

|  |            |
|--|------------|
| <b>Lecture 52: Bitwise Operators</b>   | <b>121</b> |
| <ul style="list-style-type: none"> <li>• The utility of &amp; (Bitwise AND) operator</li> <li>• To change attributes of a file using bitwise operators</li> <li>• To create function pointers</li> </ul>                                       |            |
| <b>Lecture 53: Hardware Interaction I</b>  | <b>123</b> |
| <ul style="list-style-type: none"> <li>• Ways to interact with hardware</li> <li>• Hardware interaction, DOS perspective</li> <li>• Hardware interaction, Windows perspective</li> <li>• Ports, CMOS</li> <li>• Accessing CMOS data</li> </ul> |            |
| <b>Lecture 54: Hardware Interaction II</b>   | <b>125</b> |
| <ul style="list-style-type: none"> <li>• Serial communication using null modem</li> <li>• Communication with parallel port programmatically</li> <li>• Working of Speaker</li> <li>• Playing tunes using ports</li> </ul>                      |            |
| <b>Lecture 55: Windows I</b>   | <b>127</b> |
| <ul style="list-style-type: none"> <li>• Advantages of Windows Programming model over DOS model</li> </ul>   |            |
| <b>Lecture 56: Windows II</b>  | <b>129</b> |
| <ul style="list-style-type: none"> <li>• Creating a window</li> </ul>  |            |
| <b>Lecture 57: Windows III</b>   | <b>131</b> |
| <ul style="list-style-type: none"> <li>• Event Driven Programming Model</li> <li>• To create and display a real world window</li> </ul>  |            |
| <b>Lecture 58: Windows IV</b>  | <b>133</b> |
| <ul style="list-style-type: none"> <li>• Device Context</li> <li>• Graphics under Window</li> </ul>  |            |
| <b>Lecture 59: Windows V</b>   | <b>135</b> |
| <ul style="list-style-type: none"> <li>• Freehand drawing using mouse</li> <li>• To Capture Mouse</li> <li>• To displaying Bitmap</li> <li>• Animation</li> </ul>  |            |
| <b>Lecture 60: Linux</b>   | <b>137</b> |
| <ul style="list-style-type: none"> <li>• Introduction to Linux</li> <li>• C programming under Linux</li> <li>• Parent and Child Processes</li> <li>• Communication using Signals</li> </ul>  |            |

## Contents

|   |           |
|---|-----------|
| <b>Lecture 1: Procedural Programming</b>  | <b>01</b> |
| <ul style="list-style-type: none"> <li>• Disadvantages of Procedural Programming</li> <li>• Need for Object-Oriented Programming</li> <li>• Structure in C vs. Structures in C++</li> </ul> |           |
| <b>Lecture 2: Object Oriented Solutions</b>   | <b>03</b> |
| <ul style="list-style-type: none"> <li>• Basics of Classes and Objects</li> <li>• Accessing modifiers, data members and member functions</li> </ul>   |           |
| <b>Lecture 3: Classes in CPP</b>  | <b>05</b> |
| <ul style="list-style-type: none"> <li>• How to print data</li> </ul>   |           |
| <b>Lecture 4: this Pointer</b>  | <b>07</b> |
| <ul style="list-style-type: none"> <li>• The <b>this</b> pointer and its utility</li> <li>• Creating first console-based application in VC++ IDE</li> </ul>                                 |           |
| <b>Lecture 5: Access Specifiers and Constructors</b>  | <b>09</b> |
| <ul style="list-style-type: none"> <li>• Guaranteed initialization with Constructors</li> <li>• Constructors</li> </ul>   |           |
| <b>Lecture 6: Function Overloading</b>  | <b>11</b> |
| <ul style="list-style-type: none"> <li>• Function overloading, default arguments &amp; operator overloading</li> <li>• Overloaded constructors</li> </ul>                                   |           |
| <b>Lecture 7: Calling Constructor Explicitly</b>  | <b>15</b> |
| <ul style="list-style-type: none"> <li>• Creating and initializing array of objects</li> </ul>  |           |
| <b>Lecture 8: Which gets called?</b>  | <b>17</b> |
| <ul style="list-style-type: none"> <li>• Determine which constructor gets called</li> <li>• The overloaded assignment operator &amp; copy constructor</li> </ul>                            |           |
| <b>Lecture 9: Operator Overloading</b>  | <b>19</b> |
| <ul style="list-style-type: none"> <li>• What is operator overloading?</li> </ul>   |           |
| <b>Lecture 10: Separating Things</b>  | <b>21</b> |
| <ul style="list-style-type: none"> <li>• Nameless object</li> </ul>   |           |
| <b>Lecture 11: More Variety</b>   | <b>23</b> |
| <ul style="list-style-type: none"> <li>• The <b>explicit</b> keyword</li> <li>• <b>friend</b> functions</li> </ul>  |           |
| <b>Lecture 12: Pre, Post &amp; References</b>   | <b>25</b> |
| <ul style="list-style-type: none"> <li>• Overloading pre and post incrementation operators</li> <li>• References</li> </ul>   |           |

|   |           |
|---|-----------|
| <b>Lecture 13: Are References Necessary</b>   | <b>27</b> |
| • Dos and don'ts about references   |           |
| <b>Lecture 14: Dynamic Memory Allocation</b>  | <b>29</b> |
| • Destructors   |           |
| • Dynamic memory allocation   |           |
| • Static vs. dynamic memory allocation  |           |
| <b>Lecture 15: Nameless Objects</b>   | <b>31</b> |
| • <b>malloc()/free()</b> vs. <b>new/delete</b>  |           |
| • Avoiding memory leaks and dangling pointers   |           |
| • Allocating memory dynamically for an array  |           |
| <b>Lecture 16: Static Objects</b>   | <b>33</b> |
| • Static data members and member functions  |           |
| <b>Lecture 17: Singleton Class</b>  | <b>35</b> |
| • What is mean by Singleton Class   |           |
| <b>Lecture 18: Reuse</b>  | <b>39</b> |
| • Code reuse  |           |
| • 'Has' a (container) or 'Is' a (inheritance) relationships                             |           |
| <b>Lecture 19: Containership and Inheritance</b>  | <b>41</b> |
| • Flow of construction in inheritance   |           |
| <b>Lecture 20: Object Size</b>  | <b>45</b> |
| • <b>protected</b> access modifier  |           |
| • Private, Public & Protected Inheritance   |           |
| <b>Lecture 21: Calls Differ</b>   | <b>47</b> |
| • Difference in calling mechanism for normal member functions and constructor functions |           |
| <b>Lecture 22: Types of Inheritance</b>   | <b>51</b> |
| • The Types of inheritances—Multi-level Inheritance, Multiple Inheritance               |           |
| <b>Lecture 23: Polymorphism</b>   | <b>53</b> |
| • Polymorphism  |           |
| • Compile-time & runtime polymorphism   |           |
| <b>Lecture 24: Pure Virtual Function</b>  | <b>55</b> |
| • Virtual functions   |           |
| • Pure-virtual functions  |           |
| <b>Lecture 25: Virtual Functions and Cases</b>  | <b>57</b> |
| • VTABLE and VPTR   |           |

|   |           |
|---|-----------|
| <b>Lecture 26: Different Cases</b>                    | <b>59</b> |
| • Study different cases of which function gets called |           |
| <b>Lecture 27: Virtual Destructor</b>                 | <b>61</b> |
| • Study virtual destructor                            |           |
| <b>Lecture 28: File I/O Classes</b>                   | <b>63</b> |
| • Console I/O using functions of 'stdio.h'.           |           |
| • The hierarchy of <b>ios</b> class                   |           |
| <b>Lecture 29: Formatting Flags and Manipulators</b>  | <b>65</b> |
| • Formatting flags                                    |           |
| • Manipulators  |           |
| <b>Lecture 30: Creating Manipulators</b>              | <b>67</b> |
| • User-defined manipulator                            |           |
| <b>Lecture 31: Buffered and File I/O</b>              | <b>69</b> |
| • Buffered I/O  |           |
| • The <b>iostream</b> library                         |           |
| • File I/O with Streams                               |           |
| • File opening modes                                  |           |
| • Error flags   |           |
| <b>Lecture 32: Character and Binary I/O</b>           | <b>71</b> |
| • Character I/O                                       |           |
| • Binary I/O  |           |
| • Position a file pointer                             |           |
| • Writing data to printer                             |           |
| • C++ names for hardware devices                      |           |
| <b>Lecture 33: Error Handling</b>                     | <b>73</b> |
| • Exception handling                                  |           |
| • Handle runtime error                                |           |
| • Catch Exception                                     |           |
| <b>Lecture 34: Constructors in MI</b>                 | <b>75</b> |
| • Constructors in multiple inheritance                |           |
| • Virtual base classes                                |           |
| • Virtual destructors                                 |           |
| <b>Lecture 35: Inline Functions</b>                   | <b>77</b> |
| • Inline Functions                                    |           |
| • Usage of <b>const</b> qualifier                     |           |

|  |           |
|--|-----------|
| • The <b>mutable</b> keyword   |           |
| <b>Lecture 36: Data Conversion</b>   | <b>79</b> |
| • Data conversions   |           |
| <b>Lecture 37: Object Slicing</b>  | <b>81</b> |
| • Object slicing   |           |
| • Run Time Type Identification (RTTI)  |           |
| • The usage of <b>reinterpret_cast</b> , <b>const_cast</b> , <b>static_cast</b> , etc. |           |
| <b>Lecture 38: Class Libraries</b>   | <b>83</b> |
| • Different class libraries  |           |
| • STL  |           |
| • Containers, Iterators and Algorithms   |           |
| • Categories of containers   |           |
| <b>Lecture 39: Stack, Queue and Vector</b>   | <b>85</b> |
| • Using sequence containers  |           |
| • Using derived containers   |           |
| • Using associative containers   |           |
| • Programming <b>vector</b> , <b>stack</b> and <b>priority_queue</b>                   |           |
| <b>Lecture 40: Using List</b>  | <b>87</b> |
| • Using <b>list</b>  |           |
| <b>Lecture 41: Using Set Multiset</b>  | <b>89</b> |
| • Using <b>map/multimap</b>  |           |
| • Using <b>set/multiset</b>  |           |
| <b>Lecture 42: MFC Collection Classes Introduction</b>                                 | <b>91</b> |
| • Limitations of array and linked list   |           |
| • Various MFC collection classes   |           |
| <b>Lecture 43: First Console Application</b>   | <b>93</b> |
| • Returning a reference  |           |
| • Overloading subscript ( [ ] ) operator   |           |
| • Using <b>CUIIntArray</b> class   |           |
| <b>Lecture 44: Linked List of Strings</b>  | <b>95</b> |
| • Using <b>CStringList</b> class   |           |
| <b>Lecture 45: Map Strings</b>   | <b>97</b> |
| • Using <b>CMapStringToString</b> Class  |           |
| • Setting multiple values with a key in a map  |           |

## Contents

|   |           |
|---|-----------|
| <b>Lecture 1: Procedural Programming</b>  | <b>01</b> |
| <ul style="list-style-type: none"> <li>• Disadvantages of Procedural Programming</li> <li>• Need for Object-Oriented Programming</li> <li>• Structure in C vs. Structures in C++</li> </ul> |           |
| <b>Lecture 2: Object Oriented Solutions</b>   | <b>05</b> |
| <ul style="list-style-type: none"> <li>• Basics of Classes and Objects</li> <li>• Accessing modifiers, data members and member functions</li> </ul>   |           |
| <b>Lecture 3: Classes in CPP</b>  | <b>09</b> |
| <ul style="list-style-type: none"> <li>• How to print data</li> </ul>   |           |
| <b>Lecture 4: this Pointer</b>  | <b>11</b> |
| <ul style="list-style-type: none"> <li>• The <b>this</b> pointer and its utility</li> <li>• Creating first console-based application in VC++ IDE</li> </ul>                                 |           |
| <b>Lecture 5: Access Specifiers and Constructors</b>  | <b>13</b> |
| <ul style="list-style-type: none"> <li>• Guaranteed initialization with Constructors</li> <li>• Constructors</li> </ul>   |           |
| <b>Lecture 6: Function Overloading</b>  | <b>17</b> |
| <ul style="list-style-type: none"> <li>• Function overloading, default arguments &amp; operator overloading</li> <li>• Overloaded constructors</li> </ul>                                   |           |
| <b>Lecture 7: Calling Constructor Explicitly</b>  | <b>19</b> |
| <ul style="list-style-type: none"> <li>• Creating and initializing array of objects</li> </ul>  |           |
| <b>Lecture 8: Which gets called?</b>  | <b>21</b> |
| <ul style="list-style-type: none"> <li>• Determine which constructor gets called</li> <li>• The overloaded assignment operator &amp; copy constructor</li> </ul>                            |           |
| <b>Lecture 9: Operator Overloading</b>  | <b>25</b> |
| <ul style="list-style-type: none"> <li>• What is operator overloading?</li> </ul>   |           |
| <b>Lecture 10: Separating Things</b>  | <b>27</b> |
| <ul style="list-style-type: none"> <li>• Nameless object</li> </ul>   |           |
| <b>Lecture 11: More Variety</b>   | <b>29</b> |
| <ul style="list-style-type: none"> <li>• The <b>explicit</b> keyword</li> <li>• <b>friend</b> functions</li> </ul>  |           |
| <b>Lecture 12: Pre, Post &amp; References</b>   | <b>31</b> |
| <ul style="list-style-type: none"> <li>• Overloading pre and post incrementation operators</li> <li>• References</li> </ul>   |           |

|   |           |
|---|-----------|
| <b>Lecture 13: Are References Necessary</b>   | <b>33</b> |
| • Dos and don'ts about references   |           |
| <b>Lecture 14: Dynamic Memory Allocation</b>  | <b>35</b> |
| • Destructors   |           |
| • Dynamic memory allocation   |           |
| • Static vs. dynamic memory allocation  |           |
| <b>Lecture 15: Nameless Objects</b>   | <b>37</b> |
| • <b>malloc()/free()</b> vs. <b>new/delete</b>  |           |
| • Avoiding memory leaks and dangling pointers   |           |
| • Allocating memory dynamically for an array  |           |
| <b>Lecture 16: Static Objects</b>   | <b>39</b> |
| • Static data members and member functions  |           |
| <b>Lecture 17: Singleton Class</b>  | <b>41</b> |
| • What is mean by Singleton Class   |           |
| <b>Lecture 18: Reuse</b>  | <b>43</b> |
| • Code reuse  |           |
| • 'Has' a (container) or 'Is' a (inheritance) relationships                             |           |
| <b>Lecture 19: Containership and Inheritance</b>  | <b>45</b> |
| • Flow of construction in inheritance   |           |
| <b>Lecture 20: Object Size</b>  | <b>47</b> |
| • <b>protected</b> access modifier  |           |
| • Private, Public & Protected Inheritance   |           |
| <b>Lecture 21: Calls Differ</b>   | <b>49</b> |
| • Difference in calling mechanism for normal member functions and constructor functions |           |
| <b>Lecture 22: Types of Inheritance</b>   | <b>51</b> |
| • The Types of inheritances—Multi-level Inheritance, Multiple Inheritance               |           |
| <b>Lecture 23: Polymorphism</b>   | <b>53</b> |
| • Polymorphism  |           |
| • Compile-time & runtime polymorphism   |           |
| <b>Lecture 24: Pure Virtual Function</b>  | <b>57</b> |
| • Virtual functions   |           |
| • Pure-virtual functions  |           |
| <b>Lecture 25: Virtual Functions and Cases</b>  | <b>59</b> |
| • VTABLE and VPTR   |           |
| •   |           |

|   |           |
|---|-----------|
| <b>Lecture 26: Different Cases</b>                    | <b>63</b> |
| • Study different cases of which function gets called |           |
| <b>Lecture 27: Virtual Destructor</b>                 | <b>69</b> |
| • Study virtual destructor                            |           |
| <b>Lecture 28: File I/O Classes</b>                   | <b>71</b> |
| • Console I/O using functions of 'stdio.h'.           |           |
| • The hierarchy of <b>ios</b> class                   |           |
| <b>Lecture 29: Formatting Flags and Manipulators</b>  | <b>75</b> |
| • Formatting flags                                    |           |
| • Manipulators  |           |
| <b>Lecture 30: Creating Manipulators</b>              | <b>79</b> |
| • User-defined manipulator                            |           |
| <b>Lecture 31: Buffered and File I/O</b>              | <b>81</b> |
| • Buffered I/O  |           |
| • The <b>iostream</b> library                         |           |
| • File I/O with Streams                               |           |
| • File opening modes                                  |           |
| • Error flags   |           |
| <b>Lecture 32: Character and Binary I/O</b>           | <b>85</b> |
| • Character I/O                                       |           |
| • Binary I/O  |           |
| • Position a file pointer                             |           |
| • Writing data to printer                             |           |
| • C++ names for hardware devices                      |           |
| <b>Lecture 33: Error Handling</b>                     | <b>89</b> |
| • Exception handling                                  |           |
| • Handle runtime error                                |           |
| • Catch Exception                                     |           |
| <b>Lecture 34: Constructors in MI</b>                 | <b>93</b> |
| • Constructors in multiple inheritance                |           |
| • Virtual base classes                                |           |
| • Virtual destructors                                 |           |
| <b>Lecture 35: Inline Functions</b>                   | <b>97</b> |
| • Inline Functions                                    |           |
| • Usage of <b>const</b> qualifier                     |           |

|  |            |
|--|------------|
| • The <b>mutable</b> keyword   |            |
| <b>Lecture 36: Data Conversion</b>   | <b>101</b> |
| • Data conversions   |            |
| <b>Lecture 37: Object Slicing</b>  | <b>105</b> |
| • Object slicing   |            |
| • Run Time Type Identification (RTTI)  |            |
| • The usage of <b>reinterpret_cast</b> , <b>const_cast</b> , <b>static_cast</b> , etc. |            |
| <b>Lecture 38: Class Libraries</b>   | <b>109</b> |
| • Different class libraries  |            |
| • STL  |            |
| • Containers, Iterators and Algorithms   |            |
| • Categories of containers   |            |
| <b>Lecture 39: Stack, Queue and Vector</b>   | <b>113</b> |
| • Using sequence containers  |            |
| • Using derived containers   |            |
| • Using associative containers   |            |
| • Programming <b>vector</b> , <b>stack</b> and <b>priority_queue</b>                   |            |
| <b>Lecture 40: Using List</b>  | <b>117</b> |
| • Using <b>list</b>  |            |
| <b>Lecture 41: Using Set Multiset</b>  | <b>119</b> |
| • Using <b>map/multimap</b>  |            |
| • Using <b>set/multiset</b>  |            |
| <b>Lecture 42: MFC Collection Classes Introduction</b>                                 | <b>123</b> |
| • Limitations of array and linked list   |            |
| • Various MFC collection classes   |            |
| <b>Lecture 43: First Console Application</b>   | <b>125</b> |
| • Returning a reference  |            |
| • Overloading subscript ( <code>[ ]</code> ) operator                                  |            |
| • Using <b>CUIntArray</b> class  |            |
| <b>Lecture 44: Linked List of Strings</b>  | <b>129</b> |
| • Using <b>CStringList</b> class   |            |
| <b>Lecture 45: Map Strings</b>   | <b>131</b> |
| • Using <b>CMapStringToString</b> Class  |            |
| • Setting multiple values with a key in a map  |            |

## Contents

|   |           |
|---|-----------|
| <b>Lecture 1: Procedural Programming</b>  | <b>01</b> |
| <ul style="list-style-type: none"> <li>• Disadvantages of Procedural Programming</li> <li>• Need for Object-Oriented Programming</li> <li>• Structure in C vs. Structures in C++</li> </ul> |           |
| <b>Lecture 2: Object Oriented Solutions</b>   | <b>03</b> |
| <ul style="list-style-type: none"> <li>• Basics of Classes and Objects</li> <li>• Accessing modifiers, data members and member functions</li> </ul>   |           |
| <b>Lecture 3: Classes in CPP</b>  | <b>05</b> |
| <ul style="list-style-type: none"> <li>• How to print data</li> </ul>   |           |
| <b>Lecture 4: this Pointer</b>  | <b>07</b> |
| <ul style="list-style-type: none"> <li>• The <b>this</b> pointer and its utility</li> <li>• Creating first console-based application in VC++ IDE</li> </ul>                                 |           |
| <b>Lecture 5: Access Specifiers and Constructors</b>  | <b>09</b> |
| <ul style="list-style-type: none"> <li>• Guaranteed initialization with Constructors</li> <li>• Constructors</li> </ul>   |           |
| <b>Lecture 6: Function Overloading</b>  | <b>11</b> |
| <ul style="list-style-type: none"> <li>• Function overloading, default arguments &amp; operator overloading</li> <li>• Overloaded constructors</li> </ul>                                   |           |
| <b>Lecture 7: Calling Constructor Explicitly</b>  | <b>15</b> |
| <ul style="list-style-type: none"> <li>• Creating and initializing array of objects</li> </ul>  |           |
| <b>Lecture 8: Which gets called?</b>  | <b>17</b> |
| <ul style="list-style-type: none"> <li>• Determine which constructor gets called</li> <li>• The overloaded assignment operator &amp; copy constructor</li> </ul>                            |           |
| <b>Lecture 9: Operator Overloading</b>  | <b>19</b> |
| <ul style="list-style-type: none"> <li>• What is operator overloading?</li> </ul>   |           |
| <b>Lecture 10: Separating Things</b>  | <b>21</b> |
| <ul style="list-style-type: none"> <li>• Nameless object</li> </ul>   |           |
| <b>Lecture 11: More Variety</b>   | <b>23</b> |
| <ul style="list-style-type: none"> <li>• The <b>explicit</b> keyword</li> <li>• <b>friend</b> functions</li> </ul>  |           |
| <b>Lecture 12: Pre, Post &amp; References</b>   | <b>25</b> |
| <ul style="list-style-type: none"> <li>• Overloading pre and post incrementation operators</li> <li>• References</li> </ul>   |           |

|   |           |
|---|-----------|
| <b>Lecture 13: Are References Necessary</b>   | <b>27</b> |
| • Dos and don'ts about references   |           |
| <b>Lecture 14: Dynamic Memory Allocation</b>  | <b>29</b> |
| • Destructors   |           |
| • Dynamic memory allocation   |           |
| • Static vs. dynamic memory allocation  |           |
| <b>Lecture 15: Nameless Objects</b>   | <b>31</b> |
| • <b>malloc()/free()</b> vs. <b>new/delete</b>  |           |
| • Avoiding memory leaks and dangling pointers   |           |
| • Allocating memory dynamically for an array  |           |
| <b>Lecture 16: Static Objects</b>   | <b>33</b> |
| • Static data members and member functions  |           |
| <b>Lecture 17: Singleton Class</b>  | <b>35</b> |
| • What is mean by Singleton Class   |           |
| <b>Lecture 18: Reuse</b>  | <b>39</b> |
| • Code reuse  |           |
| • 'Has' a (container) or 'Is' a (inheritance) relationships                             |           |
| <b>Lecture 19: Containership and Inheritance</b>  | <b>41</b> |
| • Flow of construction in inheritance   |           |
| <b>Lecture 20: Object Size</b>  | <b>45</b> |
| • <b>protected</b> access modifier  |           |
| • Private, Public & Protected Inheritance   |           |
| <b>Lecture 21: Calls Differ</b>   | <b>47</b> |
| • Difference in calling mechanism for normal member functions and constructor functions |           |
| <b>Lecture 22: Types of Inheritance</b>   | <b>51</b> |
| • The Types of inheritances—Multi-level Inheritance, Multiple Inheritance               |           |
| <b>Lecture 23: Polymorphism</b>   | <b>53</b> |
| • Polymorphism  |           |
| • Compile-time & runtime polymorphism   |           |
| <b>Lecture 24: Pure Virtual Function</b>  | <b>55</b> |
| • Virtual functions   |           |
| • Pure-virtual functions  |           |
| <b>Lecture 25: Virtual Functions and Cases</b>  | <b>57</b> |
| • VTABLE and VPTR   |           |

|   |           |
|---|-----------|
| <b>Lecture 26: Different Cases</b>                    | <b>59</b> |
| • Study different cases of which function gets called |           |
| <b>Lecture 27: Virtual Destructor</b>                 | <b>61</b> |
| • Study virtual destructor                            |           |
| <b>Lecture 28: File I/O Classes</b>                   | <b>63</b> |
| • Console I/O using functions of 'stdio.h'.           |           |
| • The hierarchy of <b>ios</b> class                   |           |
| <b>Lecture 29: Formatting Flags and Manipulators</b>  | <b>65</b> |
| • Formatting flags                                    |           |
| • Manipulators  |           |
| <b>Lecture 30: Creating Manipulators</b>              | <b>67</b> |
| • User-defined manipulator                            |           |
| <b>Lecture 31: Buffered and File I/O</b>              | <b>69</b> |
| • Buffered I/O  |           |
| • The <b>iostream</b> library                         |           |
| • File I/O with Streams                               |           |
| • File opening modes                                  |           |
| • Error flags   |           |
| <b>Lecture 32: Character and Binary I/O</b>           | <b>71</b> |
| • Character I/O                                       |           |
| • Binary I/O  |           |
| • Position a file pointer                             |           |
| • Writing data to printer                             |           |
| • C++ names for hardware devices                      |           |
| <b>Lecture 33: Error Handling</b>                     | <b>73</b> |
| • Exception handling                                  |           |
| • Handle runtime error                                |           |
| • Catch Exception                                     |           |
| <b>Lecture 34: Constructors in MI</b>                 | <b>75</b> |
| • Constructors in multiple inheritance                |           |
| • Virtual base classes                                |           |
| • Virtual destructors                                 |           |
| <b>Lecture 35: Inline Functions</b>                   | <b>77</b> |
| • Inline Functions                                    |           |
| • Usage of <b>const</b> qualifier                     |           |

|  |           |
|--|-----------|
| • The <b>mutable</b> keyword   |           |
| <b>Lecture 36: Data Conversion</b>   | <b>79</b> |
| • Data conversions   |           |
| <b>Lecture 37: Object Slicing</b>  | <b>81</b> |
| • Object slicing   |           |
| • Run Time Type Identification (RTTI)  |           |
| • The usage of <b>reinterpret_cast</b> , <b>const_cast</b> , <b>static_cast</b> , etc. |           |
| <b>Lecture 38: Class Libraries</b>   | <b>83</b> |
| • Different class libraries  |           |
| • STL  |           |
| • Containers, Iterators and Algorithms   |           |
| • Categories of containers   |           |
| <b>Lecture 39: Stack, Queue and Vector</b>   | <b>85</b> |
| • Using sequence containers  |           |
| • Using derived containers   |           |
| • Using associative containers   |           |
| • Programming <b>vector</b> , <b>stack</b> and <b>priority_queue</b>                   |           |
| <b>Lecture 40: Using List</b>  | <b>87</b> |
| • Using <b>list</b>  |           |
| <b>Lecture 41: Using Set Multiset</b>  | <b>89</b> |
| • Using <b>map/multimap</b>  |           |
| • Using <b>set/multiset</b>  |           |
| <b>Lecture 42: MFC Collection Classes Introduction</b>                                 | <b>91</b> |
| • Limitations of array and linked list   |           |
| • Various MFC collection classes   |           |
| <b>Lecture 43: First Console Application</b>   | <b>93</b> |
| • Returning a reference  |           |
| • Overloading subscript ( <code>[]</code> ) operator                                   |           |
| • Using <b>CUIntArray</b> class  |           |
| <b>Lecture 44: Linked List of Strings</b>  | <b>95</b> |
| • Using <b>CStringList</b> class   |           |
| <b>Lecture 45: Map Strings</b>   | <b>97</b> |
| • Using <b>CMapStringToString</b> Class  |           |
| • Setting multiple values with a key in a map  |           |

# Mathematics for Computers

Writer: Mrs Ratnavali Tilak

## Index

|         |                                 |     |
|---------|---------------------------------|-----|
| Unit 1  | Relevance of Mathematics        | 1   |
| Unit 2  | Set Theory                      | 3   |
| Unit 3  | Mathematical Induction          | 14  |
| Unit 4  | Exponents and Logarithms, Surds | 19  |
| Unit 5  | Number Systems                  | 30  |
| Unit 6  | Permutations and Combinations   | 42  |
| Unit 7  | Mathematical Logic              | 50  |
| Unit 8  | Relations                       | 62  |
| Unit 9  | Functions                       | 75  |
| Unit 10 | Vectors                         | 83  |
| Unit 11 | Matrices and Determinants       | 91  |
| Unit 12 | Mensuration                     | 104 |
| Unit 13 | System of Linear Equations      | 110 |
| Unit 14 | Polynomials                     | 119 |
| Unit 15 | Introduction to Graph theory    | 129 |



Yashwantrao  
Chavan  
Maharashtra  
Open University

**EVS 204**

## **Environmental Studies**

# **Environmental Studies**

|  |            |
|--|------------|
| <b>UNIT 1: The Multidisciplinary Nature of Environmental Studies</b> | <b>1</b>   |
| <b>UNIT 2: Natural Resources</b>                                     | <b>15</b>  |
| <b>UNIT 3: Ecosystems</b>  | <b>53</b>  |
| <b>UNIT 4: Biodiversity</b>  | <b>81</b>  |
| <b>UNIT 5: Pollution</b>   | <b>111</b> |
| <b>UNIT 6: Social Issues and the Environment</b>                     | <b>163</b> |
| <b>UNIT 7: Human Population and the Environment</b>                  | <b>213</b> |



**Yashwantrao Chavan  
Maharashtra Open  
University**

**SEC511/ OPN272-T01  
Financial and Investment  
Skills**

### **Brief Contents**

|   |            |
|---|------------|
| <b>Vice Chancellor's Message.....</b>                           | <b>3</b>   |
| <b>Forward By The Director .....</b>                            | <b>4</b>   |
| <b>Credit 01 .....</b>  | <b>5</b>   |
| <b>Credit 02.....</b>   | <b>62</b>  |
| <b>Credit 03.....</b>   | <b>203</b> |
| <b>Unit 03-01: Technical Analysis .....</b>                     | <b>204</b> |
| <b>Unit 03-02: Candlestick Patterns .....</b>                   | <b>212</b> |
| <b>Unit 03-03: Moving Average and Other Tools.....</b>          | <b>229</b> |
| <b>Credit 04.....</b>   | <b>237</b> |
| <b>Unit 04-01: Market and Taxation .....</b>                    | <b>238</b> |
| <b>Unit 04-02: Trading Psychology and Risk Management .....</b> | <b>243</b> |
| <b>Unit 04-03: Mutual Funds .....</b>                           | <b>253</b> |
| <b>Unit 04-04: Financial Planning .....</b>                     | <b>261</b> |
| <b>Answers to Self-Tests.....</b>                               | <b>263</b> |

Bachelor of Education

COURSE ON CRITICAL UNDERSTANDING OF ICT

Through Blended Learning Approach

Commonwealth of Learning, Canada& Karnataka State Open University, Mysore

# TABLE OF CONTENTS

## ACKNOWLEDGEMENT

## INTRODUCTION

## HOW TO USE THIS COURSE MATERIAL

### BLOCK-1 BASICS IN ICT AND COMPUTER APPLICATIONS

|        |  |  |
|--------|--|--|
| Unit 1 | Information and Communication Technology |  |
| Unit 2 | Computer Hardware Fundamentals           |  |
| Unit 3 | Computer Software                        |  |
| Unit 4 | Application Software                     |  |
| Unit 5 | Network and Internet                     |  |
| Unit 6 | Multimedia                               |  |

### BLOCK-2 ICT INTEGRATION IN TEACHING LEARNING PROCESS

|         |  |  |
|---------|--|--|
| Unit 7  | E- learning                              |  |
| Unit 8  | ICT Integrated Lesson                    |  |
| Unit 9  | ICT in Assessment                        |  |
| Unit 10 | ICT for educational management           |  |
| Unit 11 | ICT for Professional Development         |  |
| Unit 12 | E-Content and Open Educational Resources |  |

## WEB TECHNOLOGIES CMP508

### 1. Introduction to the Web 3 Counseling Sessions

History and Evolution  
Web development cycle Web publishing  
Web contents  
Dynamic Web contents

### 2. Languages and technologies for browsers 3 Counseling Sessions

HTML, DHTML, XHTML, ASP, JavaScript Features and Applications

### 3. Introduction to HTML 3 Counseling Sessions

HTML Fundamentals HTML Browsers  
HTML tags, Elements and Attributes Structure of HTML code  
    Head  
    Body  
Lists  
    Ordered List  
    Unordered List  
    Definition List  
    Nesting List  
Block Level Tags  
    Block formatting, Heading, Paragraph, Comments, Text alignment,  
Font size  
Text Level Tags  
    Bold, Italic, Underlined, Strikethrough, Subscript, superscript  
Inserting graphics, Scaling images Frameset  
Forms  
An introduction to DHTML, DOM

### 4. Cascading Style Sheets 3 Counseling Sessions

The usefulness of style sheets Creating style sheets  
Common tasks with CSS Font Family  
    Font Metrics  
    Units  
Properties  
Classes and Pseudo classes CSS tags

### 5. Introduction to Client side Scripting 2 Counseling Sessions

What is Scripting Language ?  
Client side and server side scripting

Types of scripting languages

## 6. JavaScript

5 Counseling Sessions

Introduction

Operators, Assignments and Comparisons, Reserved words Starting with JavaScript

Writing first JavaScript program

Putting Comments Functions

Statements in JavaScript Working with objects

Object Types and Object Instantiation

Date object, Math Object, String object, Event object, Frame object,

Screen object

Handling Events

Event handling attributes

Window Events, Form Events

Event Object

Event Simulation

## 7. XML

4 Counseling Sessions

Introduction to XML,

Anatomy of an XML document

Creating XML Documents,

Creating XML DTDs, XML Schemas, XSL

## 8. Website Design Concepts

4 Counseling Sessions

How the website should be

Basic rules of Web Page design

Types of Website

### Reference Books :

Web Technologies Achyut S. Godbole, AtulKahate Tata McGraw Hill

Web Tech. & Design C. Xavier New Age

Multimedia & Web Technology – Ramesh Bangia

HTML : The complete reference – Thomas A. Powel

JavaScript Bible – Danny Goodman