

## **ESSENTIALS OF COMPUTING**

---

# **C O N T E N T S**

### **UNIT I : COMPUTERS**

|       |                                |             |
|-------|--------------------------------|-------------|
| 1.1   | WHAT IS A COMPUTER?            | 1.1         |
| 1.2   | CHARACTERISTICS OF COMPUTERS   | 1.16        |
| 1.3   | HISTORY OF COMPUTERS           | 1.20        |
| 1.4   | CLASSIFICATION OF COMPUTERS    | 1.26        |
| 1.5   | APPLICATIONS OF COMPUTERS      | 1.29        |
| 1.6   | BASIC ORGANIZATION OF COMPUTER | 1.32        |
| 1.7   | DATA REPRESENTATION            | 1.55        |
| 1.7.1 | Introduction                   | 1.55        |
| 1.7.2 | Basics of Spreadsheet          | 1.57        |
| 1.7.3 | Data Visualization in Excel    | 1.85        |
|       | <b>ACTIVITIES</b>              | <b>1.86</b> |

### **UNIT II : COMPUTATIONAL THINKING**

|       |                                                |      |
|-------|------------------------------------------------|------|
| 2.1   | WHAT IS COMPUTATIONAL THINKING?                | 2.1  |
| 2.2   | DECOMPOSITION                                  | 2.7  |
| 2.3   | ABSTRACTION: CLASS DIAGRAMS                    | 2.10 |
| 2.3.1 | Abstraction: Use Case Diagrams                 | 2.13 |
| 2.4   | NUMBER SYSTEMS                                 | 2.16 |
| 2.5   | REAL WORLD INFORMATION TO COMPUTABLE DATA      | 2.39 |
| 2.5.1 | Converting Information into Data               | 2.43 |
| 2.2.2 | Data Capacity                                  | 2.46 |
| 2.6   | LOGIC                                          | 2.49 |
| 2.6.1 | What is Logic?                                 | 2.49 |
| 2.6.2 | Key Aspects of Logic in Computational Thinking | 2.50 |
| 2.6.3 | Types of Logic – Elaborate Notes               | 2.51 |
| 2.6.4 | Importance of Logic                            | 2.56 |
| 2.6.5 | Boolean Logic                                  | 2.59 |
| 2.6.6 | Propositional Logic                            | 2.61 |

## **ESSENTIALS OF COMPUTING**

---

|                                              |                                         |             |
|----------------------------------------------|-----------------------------------------|-------------|
| 2.6.7                                        | Writing Well-Formed Propositions        | 2.62        |
| 2.6.8                                        | Evaluating Propositions                 | 2.65        |
| 2.6.8.1                                      | Conjunction (AND)                       | 2.66        |
| 2.6.8.2                                      | Disjunction (OR)                        | 2.67        |
| 2.6.8.3                                      | Implication (IMPLIES)                   | 2.68        |
| 2.6.8.4                                      | Equivalence ( $\equiv$ )                | 2.69        |
| 2.6.8.5                                      | Logical Negation (NOT)                  | 2.70        |
| 2.6.8.6                                      | Compound Propositions                   | 2.70        |
| 2.6.8.7                                      | Logical Equivalence                     | 2.73        |
| 2.6.8.8                                      | Tautologies and Contradictions          | 2.73        |
| 2.6.9                                        | Applications of Propositional Logic     | 2.74        |
| 2.6.9.1                                      | Search Queries                          | 2.74        |
| 2.6.9.2                                      | Conjunction in Search Queries           | 2.75        |
| 2.6.9.3                                      | Disjunction in Search Queries           | 2.75        |
| 2.6.9.4                                      | Negation in Search Queries              | 2.75        |
| 2.6.10                                       | Digital Logic                           | 2.75        |
| 2.6.10.1                                     | Image Compositing                       | 2.77        |
| 2.6.10.2                                     | Database Queries                        | 2.79        |
| <b>NUMBER SYSTEM (Solutions)</b>             |                                         | <b>2.79</b> |
| <br><b>UNIT III : PROBLEM SOLVING BASICS</b> |                                         |             |
| 3.1                                          | PROBLEM DEFINITION                      | 3.1         |
| 3.2                                          | LOGICAL REASONING                       | 3.8         |
| 3.3                                          | DECOMPOSITION: SOFTWARE DESIGN          | 3.17        |
| 3.4                                          | SOFTWARE DESIGN PROCESS                 | 3.24        |
| 3.4.1                                        | What is Software Design Process?        | 3.24        |
| 3.4.2                                        | Software Design Levels                  | 3.24        |
| 3.4.3                                        | Software—Design-Phases                  | 3.26        |
| 3.4.4                                        | Elements of Software Design             | 3.27        |
| 3.4.5                                        | How Software Design Fits into the SDLC? | 3.28        |
| 3.4.6                                        | Software Design Principles              | 3.28        |
| 3.4.7                                        | Tools for Software Design               | 3.29        |
| 3.4.8                                        | Coupling and Cohesion                   | 3.29        |
| 3.4.8.1                                      | Cohesion                                | 3.30        |
| 3.4.8.2                                      | Coupling                                | 3.30        |

## **ESSENTIALS OF COMPUTING**

---

|          |                                           |       |
|----------|-------------------------------------------|-------|
| 3.4.9    | Software Analysis & Design Tools          | 3.31  |
| 3.4.9.1  | Data Flow Diagram                         | 3.31  |
| 3.4.9.2  | Types of DFD                              | 3.31  |
| 3.4.9.3  | Structure Charts                          | 3.33  |
| 3.4.9.4  | HIPO Diagram                              | 3.36  |
| 3.4.9.5  | Structured English                        | 3.37  |
| 3.4.9.6  | Pseudo-Code                               | 3.38  |
| 3.4.9.7  | Decision Tables                           | 3.39  |
| 3.4.9.10 | Entity-Relationship Model                 | 3.39  |
| 3.5      | THE CONCEPT OF AN ALGORITHM               | 3.42  |
| 3.6      | REPRESENTATION OF ALGORITHM               | 3.50  |
| 3.6.1    | VERBAL representations of an algorithm    | 3.51  |
| 3.6.2    | Pseudocode Representation of an Algorithm | 3.51  |
| 3.6.3    | Flow Chart Representation of an Algorithm | 3.54  |
| 3.7      | ALGORITHM DISCOVERY                       | 3.66  |
| 3.8      | ITERATION STATEMENTS IN PROGRAMMING       | 3.71  |
| 3.9      | RECURSIVE STRUCTURES                      | 3.79  |
| 3.9.1    | Binary Search Algorithm                   | 3.82  |
| 3.9.2    | Dynamic Programming (Top-Down)            | 3.87  |
| 3.10     | EFFICIENCY AND CORRECTNESS                | 3.95  |
| 3.10.1   | Algorithm Efficiency                      | 3.95  |
| 3.10.1.1 | Time Efficiency                           | 3.95  |
| 3.10.1.2 | Space Efficiency                          | 3.96  |
| 3.10.2   | Algorithm Correctness                     | 3.99  |
| 3.11     | IMPLEMENTATION OF ALGORITHMS              | 3.105 |
| 3.11.1   | Phases of Implementation                  | 3.105 |
| 3.11.2   | Methods of Implementation                 | 3.107 |
| 3.11.3   | Fundamental Algorithms                    | 3.111 |
| 3.11.3.1 | Exchanging the values of two variables    | 3.111 |
| 3.11.3.2 | Counting of Numbers                       | 3.113 |
| 3.11.3.3 | Summation of a set of numbers             | 3.117 |
| 3.11.3.4 | Factorial Computation                     | 3.120 |
| 3.11.3.5 | Generation of Fibonacci Sequence          | 3.122 |

## **ESSENTIALS OF COMPUTING**

---

|                                             |              |
|---------------------------------------------|--------------|
| 3.11.3.6 Reversing the digits of an Integer | 3.124        |
| 3.11.3.7 Base Conversion                    | 3.126        |
| <b>ACTIVITIES</b>                           | <b>3.129</b> |

## **UNIT IV : PROGRAMMING LANGUAGES**

|                                                        |      |
|--------------------------------------------------------|------|
| 4.1 PROGRAM DEVELOPMENT LIFE CYCLE                     | 4.1  |
| 4.2 PROGRAM DESIGN TOOLS                               | 4.5  |
| 4.2.1 What are Software Analysis and Design Tools?     | 4.7  |
| 4.2.2 Data Flow Diagram                                | 4.7  |
| 4.2.3 Structure Charts                                 | 4.13 |
| 4.2.4 HIPO Diagram                                     | 4.16 |
| 4.2.5 Structured English                               | 4.17 |
| 4.2.6 Pseudo-Code                                      | 4.17 |
| 4.2.7 Decision Tables                                  | 4.18 |
| 4.2.8 Entity-Relationship Model                        | 4.22 |
| 4.2.9 Data Dictionary                                  | 4.32 |
| 4.3 ALGORITHMS                                         | 4.36 |
| 4.3.1 How to Design an Algorithm?                      | 4.40 |
| 4.3.2 How to Analyze an Algorithm?                     | 4.42 |
| 4.3.3 What is Algorithm complexity and how to find it? | 4.43 |
| 4.3.4 How to Calculate Space Complexity?               | 4.43 |
| 4.3.5 How to Calculate, Time Complexity?               | 4.44 |
| 4.3.6 How to express an Algorithm?                     | 4.45 |
| 4.4 FLOWCHARTS                                         | 4.45 |
| 4.4.1 When to Use Flowchart?                           | 4.47 |
| 4.5 PSEUDOCODES                                        | 4.55 |
| 4.5.1 How to write a Pseudo-code?                      | 4.55 |
| 4.6 ROLE OF ALGORITHMS                                 | 4.56 |
| 4.7 PROGRAMMING LANGUAGES                              | 4.60 |
| 4.8 PROGRAMMING PARADIGMS                              | 4.63 |
| 4.9 TRADITIONAL PROGRAMMING CONCEPTS                   | 4.69 |
| 4.10 PROCEDURAL UNITS                                  | 4.72 |

## **ESSENTIALS OF COMPUTING**

---

|                              |      |
|------------------------------|------|
| 4.11 LANGUAGE IMPLEMENTATION | 4.75 |
| 4.12 DECLARATIVE PROGRAMMING | 4.78 |
| 4.13 SUGGESTED ACTIVITIES    | 4.80 |

## **UNIT V : SCRATCH PROGRAMMING**

|                                                    |      |
|----------------------------------------------------|------|
| 5.1 WHAT IS SCRATCH?                               | 5.1  |
| 5.2 SCRATCH PROGRAMMING ENVIRONMENT                | 5.3  |
| 5.3 PAINT EDITOR                                   | 5.16 |
| 5.3.1 Example-1: First scratch game                | 5.17 |
| 5.4 SCRATCH BLOCKS                                 | 5.22 |
| 5.5 ARITHMETIC OPERATORS AND FUNCTIONS             | 5.23 |
| 5.5.1 Arithmetic Operators                         | 5.23 |
| 5.6 MOTION AND DRAWING                             | 5.24 |
| 5.6.1 Using Motion Commands                        | 5.24 |
| 5.7 PEN COMMANDS AND EASY DRAW                     | 5.31 |
| 5.8 THE POWER OF REPEAT                            | 5.50 |
| 5.9 LOOKS AND SOUND                                | 5.54 |
| 5.9.1 The Looks Palette                            | 5.54 |
| 5.9.2 Changing Costumes to Animate (Animation.sb2) | 5.54 |
| 5.9.3 Sprites That Speak and Think                 | 5.56 |
| 5.9.4 Image Effects (GraphicEffects .sb2)          | 5.56 |
| 5.9.5 Size and Visibility                          | 5.57 |
| 5.9.6 Layers (Layers.sb2)                          | 5.58 |
| 5.10 THE SOUND PALETTE                             | 5.58 |
| 5.10.1 Playing Audio Files                         | 5.58 |
| 5.10.2 Playing Drums and Other Sounds              | 5.59 |
| 5.10.3 Composing Music                             | 5.60 |
| 5.10.4 Controlling Sound Volume (VolumeDemo.sb2)   | 5.61 |
| 5.10.5 Setting the Tempo                           | 5.61 |
| 5.11 VARIABLES                                     | 5.62 |
| 5.11.1 Data Types in Scratch                       | 5.62 |

## **ESSENTIALS OF COMPUTING**

---

|                   |                                                               |              |
|-------------------|---------------------------------------------------------------|--------------|
| 5.11.2            | What is in shapes?                                            | 5.62         |
| 5.11.3            | Automatic Data Type Conversion                                | 5.63         |
| 5.12              | INTRODUCTION TO VARIABLES                                     | 5.64         |
| 5.13              | GETTING INPUT FROM USERS                                      | 5.73         |
| 5.14              | MAKING DECISIONS - OPERATORS IN SCRATCH                       | 5.80         |
| 5.14.1            | Relational/Comparison Operator                                | 5.80         |
| 5.15              | LOGICAL-OPERATORS IN SCRATCH                                  | 5.83         |
| 5.16              | COMPARING LETTERS AND STRINGS                                 | 5.86         |
| 5.17              | DECISION STRUCTURES                                           | 5.88         |
| 5.18              | SCRATCH ITERATION BLOCK                                       | 5.93         |
| 5.19              | STOP COMMAND                                                  | 5.100        |
| 5.20              | COUNTERS                                                      | 5.103        |
| 5.20.1            | Check a Password- Password Check.sb2                          | 5.103        |
| 5.20.2            | Examples                                                      | 5.104        |
| 5.21              | NESTED LOOPS                                                  | 5.106        |
| 5.22              | RECURSION: PROCEDURES THAT CALL THEMSELVES -<br>RECURSION.SB2 | 5.111        |
| 5.23              | STRING PROCESSING                                             | 5.113        |
| 5.23.1            | String data Types                                             | 5.113        |
| 5.23.2            | Counting Special Characters in a String- VowelCount.sb2       | 5.113        |
| 5.23.3            | Comparing String Characters -Palindrome.sb2                   | 5.114        |
| 5.23.4            | String Manipulation                                           | 5.116        |
| 5.24              | LIST                                                          | 5.119        |
| 5.25              | DYNAMIC LISTS                                                 | 5.121        |
| 5.26              | NUMERICAL LISTS IN SCRATCH                                    | 5.125        |
| 5.27              | SORTING NUMBERS IN SCRATCH                                    | 5.129        |
| 5.28              | SEARCHING IN A LIST (LINEAR SEARCH)                           | 5.130        |
| <b>ACTIVITIES</b> |                                                               | <b>5.131</b> |

## **UNIT VI : APP DEVELOPMENT**

|     |                                                |     |
|-----|------------------------------------------------|-----|
| 6.1 | BUILDING AN APP USING A PROBLEM                | 6.1 |
| 6.2 | SOLVING TECHNIQUES ON APP DEVELOPMENT PLATFORM | 6.3 |

## **ESSENTIALS OF COMPUTING**

---

|      |                                                  |             |
|------|--------------------------------------------------|-------------|
| 6.3  | SOFTWARE DEVELOPMENT MODELS                      | 6.5         |
| 6.4  | INCREMENTAL MODEL                                | 6.12        |
| 6.5  | ITERATIVE MODEL                                  | 6.17        |
| 6.6  | REUSABLE MODULES FOR MOBILE APPS                 | 6.21        |
| 6.7  | PROBLEMS WITH REUSE                              | 6.22        |
| 6.8  | THE REUSE LANDSCAPE                              | 6.23        |
| 6.9  | APPROACH THAT SUPPORTS SOFTWARE REUSE            | 6.24        |
| 6.10 | MODULARIZATION                                   | 6.26        |
| 6.11 | ALGORITHM THINKING TECHNIQUES IN APP DEVELOPMENT | 6.28        |
| 6.12 | ABSTRACTION IN APP DEVELOPMENT                   | 6.31        |
| 6.13 | DECOMPOSITION METHOD IN APP DEVELOPMENT          | 6.33        |
| 6.14 | TESTING AND DEBUGGING                            | 6.37        |
|      | <b>ACTIVITIES</b>                                | <b>6.46</b> |

## **TWO MARKS QUESTIONS AND ANSWERS**