**Module 1** ( 8 sessions)

**Computer Appreciation**

1. BRIEF HISTORY OF COMPUTERS

a. Type of computer: Desktop, Laptop, Tablet

2. COMPUTER HARDWARE

a. Input devices

 Mouse, keyboard, scanner

b. Output devices

Printers, monitor,

3. COMPUTER SOFTWARE

OS, Browser, Word processor, Media Player, Antivirus, Spreadsheet

4. UNDERSTAND THE DESKTOP

a. Desktop screen b. Start button c. Task bar d. Notification area

5. MANAGING WINDOWS

a. Minimize/Maximize b. Scrollbar

6. PRACTICE

a. Typing b. Clicking

7. INSIDE YOUR COMPUTER

a. RAM b. Hard drive

8. BYTES

a. Calculations involving bytes & measurements

9. INTRO TO FILES & FOLDERS

a. Intro b. Open, create & save and rename files and folders

10. MENUS WITHIN A PROGRAM

a. file, edit, print

11. MOVING FILES AND DATA

a. copy & paste b. drag & drop

12. BRIEF HISTORY OF THE INTERNET

13. GETTING ONLINE

14. UNDERSTANDING LINKS

15. USING SEARCH ENGINES

16. BASIC WEBSITE NAVIGATION

17. CREATING USERNAMES & PASSWORDS

18. DOWNLOADING & UPLOADING THRU THE INTERNET

OBJECTIVES

At the end of the module, the student(s) should

1. Know types of computers

2. Identify and use computer hardware parts such as keyboard, mouse, power button, etc

3. Understand computer software parts such as Operating System, browser, antivirus, etc

4. Identify and know the uses of items found on the desktop workspace such as start button, taskbar, notification area, etc

5. Understand RAM, ROM and HDD

6. Understand bytes as the primary unit of data storage

7. Be able to distinguish between files and folders

8. Be able to open, create, rename and save files and folders

9. Understand the typical menu items in a program

10. Know brief history of the internet and how to get online

11. Use search engines and navigate a website

12. Create username and password

13. Upload and download files through the internet

**Module** **2**

**Programming Language Concepts**

1. VARIABLES

a. Declare and initialize

2. CONSTANTS & LITERALS

3. METHODS

a.declare and call

4. KEYWORDS

5. COMMENTS

6. DATA TYPES

a. integers, strings, boolean, floats

7. TYPE CONVERSION

8. OPERATORS

a. arithmetic, relational, logical, assignment, bitwise

9. ARRAYS

a. declare and initialize

**OBJECTIVES**

At the end of the module, the student(s) should

1. Understand what variables are and declare and initialize them

2. Recognize and use constants and literals

3. Know what functions are and be able to declare and call them

4. Be familiar with reserved keywords of a programming language

5. Understand and use comments

6. Understand data types and when to use them

7. Be able to convert between data types and cast

8. Understand arrays and be able to declare and initialize them

**Module 3**

**Control Flow**

1. DECISION MAKING

a. If, switch,

2. LOOPS

for, while, foreach, break, continue, infinite loop

**OBJECTIVES**

At the end of the module, the student(s) should

1. Understand and know how to construct if and switch statements

2. Understand how to construct and use for, while & foreach loops

**Module 4**

**Algorithm and Data Structures**

1. INTRO TO ALGORITHM

a. definition, xtics

2. WRITE AND ANALYZE ALGORITHMS

3. INTRO TO FLOWCHARTS

4. Searching and sorting algorithms

5. INTRO TO DATA STRUCTURES

a. built-in - integers, boolean, characters

b. derived - lists, stacks, queue, trees

**OBJECTIVES**

At the end of the module, the student(s) should

1. Know what algorithms are

2. Write and analyze algorithms

3. be familiar with common searching and sorting algorithms

4. Draw flowcharts

5. Be familiar with built-in and derived data structures such as strings, integers, lists, queues, etc

**Module 5**

**Error handling and Debugging**

1. THE NEED FOR ERROR HANDLING

2. TRY-CATCH BLOCK

3. THROW

4. FINALLY

**OBJECTIVES**

At the end of the module, the student(s) should

1. Understand why errors should be anticipated and handled

2. Use try catch exception

3. Use throw statement

4. Use finally statement

**Module 6**

**Intro to Object Oriented Programming**

1. NAMESPACES

2. CLASSES

3. INTERFACES

4. INHERITANCE

5. POLYMORPHISM

**OBJECTIVES**

At the end of the module, the student(s) should

1. Understand the concept of object oriented programming

2. Understand namespaces

3. Understand classes

4. Understand interfaces

5. Understand the concepts of inheritance and polymorphism as the tenets of OOP

**Module 7**

**Input/Output Operations**

1. READ AND WRITE FILES

**OBJECTIVES**

At the end of the module, the student(s) should

1. Be able to read files

2. Be able to write files

**Module 8**

**Application Performance & Memory Management**

1. GARBAGE COLLECTION

2.

**OBJECTIVES**

At the end of the module, the student(s) should

1. Understand garbage collection
2. Understand time complexity
3. Understand space complexity