

DIGITAL TECHNOLOGIES

SENIOR SECONDARY SCHOOL 2 (SS2)

FIRST TERM E-LESSON NOTES

WEEK 1: REVISION OF SS1 WORK

Learning Objectives:

By the end of this lesson, students should be able to:

- Recall basic networking concepts from SS1
- Identify fundamental web development elements
- Explain basic database concepts
- Discuss cybersecurity principles

Previous Knowledge:

Students have basic knowledge of computer fundamentals, simple networking, and introductory programming concepts.

Lesson Content:

A. Networks Review

- **Network Types:** LAN, WAN, MAN
- **Network Topologies:** Star, Ring, Bus, Mesh
- **Network Devices:** Router, Switch, Hub, Modem
- **Network Media:** Twisted pair, Coaxial, Fiber optic

B. Web Development Basics

- **HTML Structure:** Tags, elements, attributes
- **Basic HTML Tags:** <html>, <head>, <body>, <h1>-<h6>, <p>, <div>
- **Web Browsers:** Chrome, Firefox, Safari, Edge
- **Client-Server Model**

C. Database Fundamentals

- **Database:** Collection of organized data
- **DBMS:** Database Management System

- **Tables, Records, Fields**
- **Primary Key concept**

D. Cybersecurity Basics

- **Threats:** Virus, Malware, Phishing
- **Protection:** Antivirus, Firewalls, Strong passwords
- **Data Backup importance**

Evaluation Questions:

1. Define a computer network and state three advantages
2. List four HTML tags and their functions
3. What is a database? Give two examples
4. Mention three cybersecurity threats

Assignment:

Create a simple HTML page with your personal information using basic tags learned in SS1.

WEEK 2: COMPUTER ARCHITECTURE (ADVANCED)

Learning Objectives:

Students should be able to:

- Explain CPU design and components
- Describe memory hierarchy
- Analyze data flow in computer systems
- Identify different types of buses

Previous Knowledge:

Basic computer components and their functions

Lesson Content:

A. CPU Design and Components

- **Control Unit (CU):** Coordinates and controls computer operations
- **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations
- **Registers:** High-speed storage locations within CPU
 - Program Counter (PC)

- Memory Address Register (MAR)
- Memory Data Register (MDR)
- Accumulator

B. Memory Hierarchy

1. **Primary Memory:**
 - RAM (Random Access Memory)
 - ROM (Read Only Memory)
 - Cache Memory (L1, L2, L3)
2. **Secondary Memory:**
 - Hard Disk Drive (HDD)
 - Solid State Drive (SSD)
 - Optical Disks (CD, DVD, Blu-ray)
3. **Tertiary Storage:**
 - Tape drives
 - Cloud storage

C. System Buses

- **Data Bus:** Carries actual data between components
- **Address Bus:** Carries memory addresses
- **Control Bus:** Carries control signals

D. Data Flow Process

1. Instruction Fetch
2. Instruction Decode
3. Execute
4. Store Result (Fetch-Decode-Execute Cycle)

Practical Activity:

Draw and label a block diagram of computer architecture showing CPU, memory, and buses.

Evaluation Questions:

1. State the two main components of CPU and their functions
2. Arrange the following in order of speed: RAM, Cache, Hard disk
3. Explain the fetch-decode-execute cycle
4. What is the function of the address bus?

Assignment:

Research and write a short note on modern CPU architectures (Intel vs AMD).

WEEK 3: OPERATING SYSTEMS (FILE & PROCESS MANAGEMENT)

Learning Objectives:

Students should be able to:

- Define operating system and its functions
- Explain multitasking and process scheduling
- Understand file management systems
- Discuss file permissions and security

Previous Knowledge:

Basic knowledge of operating systems like Windows, Linux, macOS

Lesson Content:

A. Operating System Overview

- **Definition:** Software that manages computer hardware and software resources
- **Functions:**
 - Process Management
 - Memory Management
 - File Management
 - Device Management
 - Security Management

B. Process Management

- **Process:** Program in execution
- **Multitasking:** Running multiple processes simultaneously
- **Process States:**
 - New
 - Ready
 - Running
 - Waiting
 - Terminated

C. Process Scheduling

- **First Come First Serve (FCFS)**
- **Shortest Job First (SJF)**

- **Round Robin**
- **Priority Scheduling**

D. File Management

- **File System:** Method of organizing and storing files
- **Directory Structure:** Hierarchical organization
- **File Operations:** Create, Read, Write, Delete
- **File Allocation Methods:**
 - Contiguous
 - Linked
 - Indexed

E. File Permissions

- **Read (R):** View file contents
- **Write (W):** Modify file contents
- **Execute (X):** Run file as program
- **User Categories:** Owner, Group, Others

Practical Activity:

Navigate file system using command prompt/terminal and practice changing file permissions.

Evaluation Questions:

1. Define operating system and state four functions
2. What is multitasking? Give two advantages
3. Explain three process scheduling algorithms
4. What do the permissions rwx-r--r-- mean?

Assignment:

Compare file management systems of Windows and Linux operating systems.

WEEK 4: DATA STRUCTURES & ALGORITHMS (INTRODUCTION)

Learning Objectives:

Students should be able to:

- Define data structures and their importance

- Understand arrays and lists
- Create simple flowcharts
- Write basic pseudocode

Previous Knowledge:

Basic programming concepts and mathematical operations

Lesson Content:

A. Data Structures Overview

- **Definition:** Way of organizing and storing data efficiently
- **Importance:**
 - Efficient data access
 - Better memory utilization
 - Faster processing
 - Code reusability

B. Arrays

- **Definition:** Collection of elements of same data type
- **Characteristics:**
 - Fixed size
 - Elements accessed by index
 - Homogeneous data
- **Types:**
 - One-dimensional arrays
 - Multi-dimensional arrays

C. Lists

- **Definition:** Ordered collection of elements
- **Types:**
 - Linked Lists
 - Dynamic Lists
- **Operations:**
 - Insert
 - Delete
 - Search
 - Traverse

D. Flowcharts

- **Definition:** Graphical representation of algorithm
- **Symbols:**

- Oval: Start/End
- Rectangle: Process
- Diamond: Decision
- Parallelogram: Input/Output
- Circle: Connector

E. Pseudocode

- **Definition:** High-level description of algorithm
- **Structure:**
- BEGIN INPUT data PROCESS data OUTPUT resultEND

Practical Activity:

Create a flowchart and pseudocode for finding the largest number in an array.

Evaluation Questions:

1. What is a data structure? State two advantages
2. Differentiate between arrays and lists
3. Draw flowchart symbols for process and decision
4. Write pseudocode to calculate average of 5 numbers

Assignment:

Create a flowchart for a simple calculator that performs basic arithmetic operations.

WEEK 5: ADVANCED WORD PROCESSING & DESKTOP PUBLISHING

Learning Objectives:

Students should be able to:

- Design professional page layouts
- Create newsletters and brochures
- Use advanced formatting features
- Apply desktop publishing principles

Previous Knowledge:

Basic word processing skills and document formatting

Lesson Content:

A. Advanced Page Layout

- **Page Setup:**
 - Margins and orientation
 - Page size and columns
 - Headers and footers
- **Text Formatting:**
 - Typography principles
 - Font pairing
 - Line spacing and alignment
- **Paragraph Formatting:**
 - Indentation
 - Spacing
 - Borders and shading

B. Newsletter Design

- **Structure:**
 - Masthead
 - Headlines
 - Body text
 - Sidebar content
- **Design Principles:**
 - Hierarchy
 - Balance
 - Contrast
 - Alignment

C. Brochure Creation

- **Types:**
 - Bi-fold
 - Tri-fold
 - Z-fold
- **Elements:**
 - Cover design
 - Content organization
 - Contact information
 - Call-to-action

D. Desktop Publishing Principles

- **White Space:** Effective use of empty areas
- **Grid System:** Consistent alignment

- **Color Theory:** Harmonious color schemes
- **Image Integration:** Proper placement and sizing

Practical Activity:

Design a school newsletter including articles, images, and proper formatting using MS Word or similar software.

Evaluation Questions:

1. State four elements of a good newsletter design
2. What is desktop publishing?
3. Mention three types of brochure folds
4. Explain the importance of white space in design

Assignment:

Create a tri-fold brochure advertising your school's computer science program.

WEEK 6: SPREADSHEET DATA MODELLING

Learning Objectives:

Students should be able to:

- Perform scenario analysis
- Create and use pivot tables
- Design interactive dashboards
- Apply advanced Excel functions

Previous Knowledge:

Basic spreadsheet operations and formula creation

Lesson Content:

A. Scenario Analysis

- **What-If Analysis:**
 - Data Tables
 - Goal Seek
 - Solver
- **Scenarios:**

- Best case
- Worst case
- Most likely case

B. Pivot Tables

- **Purpose:** Summarize and analyze large datasets
- **Components:**
 - Row fields
 - Column fields
 - Data fields
 - Filter fields
- **Operations:**
 - Grouping data
 - Calculated fields
 - Sorting and filtering

C. Dashboard Creation

- **Elements:**
 - Charts and graphs
 - Key performance indicators (KPIs)
 - Interactive controls
 - Data visualization
- **Design Principles:**
 - Clarity
 - Relevance
 - Interactivity
 - Visual appeal

D. Advanced Functions

- **Lookup Functions:** VLOOKUP, HLOOKUP, INDEX, MATCH
- **Text Functions:** CONCATENATE, LEFT, RIGHT, MID
- **Date Functions:** TODAY, NOW, DATEDIF
- **Statistical Functions:** AVERAGE, MEDIAN, MODE

Practical Activity:

Create a sales dashboard with pivot tables showing monthly sales analysis and trend charts.

Evaluation Questions:

1. What is scenario analysis? Give two examples
2. State four components of a pivot table
3. Mention three elements of an effective dashboard

4. Write the syntax for VLOOKUP function

Assignment:

Design a student grade analysis system using pivot tables and charts.

WEEK 7: WEB DEVELOPMENT (CSS & JAVASCRIPT BASICS)

Learning Objectives:

Students should be able to:

- Apply CSS styling to HTML documents
- Understand CSS selectors and properties
- Write basic JavaScript code
- Add interactivity to web pages

Previous Knowledge:

HTML basics and web page structure

Lesson Content:

A. CSS Fundamentals

- **Definition:** Cascading Style Sheets for styling HTML
- **CSS Syntax:**
`selector { property: value; }`
- **Three Ways to Add CSS:**
 - Inline CSS
 - Internal CSS
 - External CSS

B. CSS Selectors

- **Element Selector:** `p { }`
- **Class Selector:** `.classname { }`
- **ID Selector:** `#idname { }`
- **Universal Selector:** `* { }`

C. CSS Properties

- **Text Properties:**
 - color
 - font-family
 - font-size
 - text-align
- **Box Model:**
 - margin
 - padding
 - border
 - width/height

D. JavaScript Basics

- **Definition:** Programming language for web interactivity
- **Variables:**
- `var name = "John"; let age = 25; const pi = 3.14;`
- **Functions:**
- `function greetUser() { alert("Hello, World!"); }`
- **Events:**
 - onclick
 - onload
 - onsubmit

Practical Activity:

Create a styled web page with CSS and add a simple JavaScript function for user interaction.

Evaluation Questions:

1. What is CSS? State three ways to add CSS to HTML
2. Differentiate between class and ID selectors
3. Write JavaScript code to display an alert message
4. What is the CSS box model?

Assignment:

Create a personal portfolio webpage using HTML, CSS, and basic JavaScript.

WEEK 8: DATABASE MANAGEMENT SYSTEMS (INTERMEDIATE)

Learning Objectives:

Students should be able to:

- Understand relational database concepts
- Write basic SQL queries
- Perform database operations
- Design simple database schemas

Previous Knowledge:

Basic database concepts and table structure

Lesson Content:

A. Relational Database Management System (RDBMS)

- **Definition:** Database system using relational model
- **Features:**
 - Tables with rows and columns
 - Relationships between tables
 - Data integrity
 - ACID properties

B. Database Design Concepts

- **Entity:** Object or thing in real world
- **Attribute:** Property of an entity
- **Relationship:** Connection between entities
- **Primary Key:** Unique identifier for records
- **Foreign Key:** Link between tables

C. SQL (Structured Query Language)

- **Definition:** Standard language for database operations
- **SQL Categories:**
 - DDL (Data Definition Language)
 - DML (Data Manipulation Language)
 - DCL (Data Control Language)

D. Basic SQL Commands

- **SELECT:** Retrieve data
- `SELECT * FROM students; SELECT name, age FROM students WHERE age > 18;`
- **INSERT:** Add new records
- `INSERT INTO students (name, age, grade) VALUES ('John Doe', 17, 'SS2');`
- **UPDATE:** Modify existing records
- `UPDATE students SET grade = 'SS3' WHERE name = 'John Doe';`

- **DELETE:** Remove records
- `DELETE FROM students WHERE age < 16;`

Practical Activity:

Create a simple student database and practice SQL commands using a database management tool.

Evaluation Questions:

1. What is RDBMS? State three features
2. Differentiate between primary key and foreign key
3. Write SQL command to select all records from a table named "teachers"
4. What does UPDATE command do in SQL?

Assignment:

Design a database schema for a school library system with appropriate tables and relationships.

WEEK 9: PYTHON PROGRAMMING (INTERMEDIATE)

Learning Objectives:

Students should be able to:

- Define and use functions in Python
- Implement error handling
- Develop simple Python projects
- Apply programming best practices

Previous Knowledge:

Basic Python syntax, variables, and control structures

Lesson Content:

A. Functions in Python

- **Definition:** Reusable blocks of code
- **Function Definition:**
- `def function_name(parameters):` `# function body` `return value`
- **Example:**

- `def calculate_area(length, width):      area = length * width      return area`
- **Function Types:**
 - Built-in functions
 - User-defined functions
 - Lambda functions

B. Parameters and Arguments

- **Positional Arguments**
- **Keyword Arguments**
- **Default Parameters:**
- `def greet(name, message="Hello"):      print(f"{message}, {name}!")`
- ****Variable-length Arguments (*args, kwargs)**

C. Error Handling

- **Exception Handling:**
- `try:      result = 10 / 0except ZeroDivisionError:      print("Cannot divide by zero!")finally:      print("Execution completed")`
- **Common Exceptions:**
 - ValueError
 - TypeError
 - IndexError
 - KeyError

D. Simple Projects

- **Calculator Program**
- **Grade Management System**
- **Number Guessing Game**
- **Simple Quiz Application**

Practical Activity:

Create a simple calculator program with functions for basic arithmetic operations and error handling.

Code Example:

```
def calculator():
    def add(x, y):
        return x + y

    def subtract(x, y):
        return x - y

    def multiply(x, y):
```

```

        return x * y

def divide(x, y):
    try:
        return x / y
    except ZeroDivisionError:
        return "Error: Cannot divide by zero"

print("Simple Calculator")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")

choice = input("Select operation (1-4): ")

try:
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))

    if choice == '1':
        print(f"Result: {add(num1, num2)}")
    elif choice == '2':
        print(f"Result: {subtract(num1, num2)}")
    elif choice == '3':
        print(f"Result: {multiply(num1, num2)}")
    elif choice == '4':
        print(f"Result: {divide(num1, num2)}")
    else:
        print("Invalid choice!")

except ValueError:
    print("Invalid input! Please enter numbers only.")

calculator()

```

Evaluation Questions:

1. What is a function? State two advantages of using functions
2. Write a Python function to calculate the square of a number
3. What is the purpose of try-except blocks?
4. How do you pass default parameters to a function?

Assignment:

Create a student grade calculator that takes multiple subject scores and calculates GPA with appropriate error handling.

WEEK 10-11: ICT & SOCIETY: ETHICAL AND LEGAL ISSUES

Learning Objectives:

Students should be able to:

- Understand digital ethics and responsibilities
- Explain copyright and intellectual property laws
- Discuss data protection principles
- Analyze the social impact of ICT

Previous Knowledge:

Basic understanding of ICT applications in society

Lesson Content:

A. Digital Ethics

- **Definition:** Moral principles governing digital behavior
- **Key Areas:**
 - Privacy rights
 - Digital divide
 - Algorithmic bias
 - Artificial intelligence ethics

B. Digital Law and Regulations

- **Computer Crimes:**
 - Hacking and unauthorized access
 - Identity theft
 - Cyberbullying
 - Online fraud
- **Legal Frameworks:**
 - Computer Fraud and Abuse Act
 - General Data Protection Regulation (GDPR)
 - Digital Millennium Copyright Act (DMCA)

C. Copyright and Intellectual Property

- **Copyright:** Legal protection for creative works
- **Types of IP:**
 - Patents
 - Trademarks

- Trade secrets
- Copyrights
- **Fair Use Doctrine**
- **Creative Commons Licensing**

D. Data Protection and Privacy

- **Personal Data:** Information that identifies individuals
- **Data Protection Principles:**
 - Lawfulness and transparency
 - Purpose limitation
 - Data minimization
 - Accuracy
 - Storage limitation
 - Security
 - Accountability

E. Social Impact of ICT

- **Positive Impacts:**
 - Global connectivity
 - Access to information
 - Economic opportunities
 - Educational advancement
- **Negative Impacts:**
 - Digital addiction
 - Job displacement
 - Privacy concerns
 - Cybersecurity threats

Case Study Activity:

Analyze a real-world case of digital rights violation and discuss the ethical and legal implications.

Evaluation Questions:

1. Define digital ethics and state three key areas
2. What is copyright? How does it protect creators?
3. List four principles of data protection
4. Discuss two positive and two negative impacts of ICT on society

Assignment:

Write an essay on "The Role of Digital Ethics in Modern Society" (500 words).

WEEK 12: REVISION & EXAMINATION

Learning Objectives:

Students should be able to:

- Review all topics covered in the term
- Apply knowledge to solve problems
- Prepare effectively for examinations

Revision Topics:

A. Key Concepts Review

1. **Computer Architecture:** CPU, memory hierarchy, buses
2. **Operating Systems:** Process management, file systems
3. **Data Structures:** Arrays, lists, algorithms
4. **Web Development:** HTML, CSS, JavaScript basics
5. **Database Systems:** SQL, RDBMS concepts
6. **Python Programming:** Functions, error handling
7. **ICT Ethics:** Digital law, privacy, copyright

B. Problem-Solving Practice

- Algorithm development
- Database query writing
- Web page creation
- Python programming exercises

C. Examination Preparation

- **Question Types:**
 - Multiple choice
 - Short answer
 - Essay questions
 - Practical demonstrations
- **Study Strategies:**
 - Active recall
 - Spaced repetition
 - Practice tests
 - Group study

Sample Examination Questions:

Section A: Multiple Choice (20 marks)

1. Which component of CPU performs arithmetic operations? a) Control Unit b) ALU c) Register d) Cache
2. In CSS, which selector is used for ID? a) . b) # c) * d) %

Section B: Short Answer (30 marks)

1. Explain the fetch-decode-execute cycle (5 marks)
2. Write SQL command to create a table named "books" with columns: id, title, author, year (5 marks)

Section C: Essay Questions (50 marks)

1. Discuss the importance of data protection in the digital age (15 marks)
2. Compare and contrast arrays and linked lists as data structures (15 marks)

Practical Assessment:

Students will be required to:

- Create a simple webpage with HTML, CSS, and JavaScript
- Write a Python program to solve a given problem
- Demonstrate database operations using SQL

RECOMMENDED RESOURCES

Textbooks:

1. "New School Digital Technologies for Senior Secondary Schools" by Hiit Plc
2. "Computer Studies for Senior Secondary Schools" by M.E. Adegbija
3. "Digital Technologies Made Simple" by Tech Education Publishers