BUDHA DAL PUBLIC SCHOOL, PATIALA

LESSON PLAN OF CLASS XI (SUBJECT: INFORMATICS PRACTICES 065)

Term –I & Final Exams Syllabus (Session 2025-26)

Month wise Distribution

April

Ch-1 Computer System

May

Ch-2 Getting Started with Python

Ch-3 Python Programming Fundamental

JULY

Ch-4 Conditional and Looping Constructs

AUGUST

Ch-5 Lists in Python

SEPTEMBER

Revision + Half yearly exam

OCTOBER

Ch-6 Dictionary

Ch-7 Database Concepts

NOVEMBER

Ch-8 Structured Query Language

DECEMBER /JANUARY

Final Practical & Revision of Final Examination

FEBRUARY/MARCH

FINAL EXAMS

PERIODIC-I SYLLABUS

Ch-1 Computer System

Ch-2 Getting Started with Python

TERM-I SYLLABUS

Ch-1 Computer System

Ch-2 Getting Started with Python

Ch-3 Python Programming Fundamental

Ch-4 Conditional and Looping Constructs

Ch-5 Lists in Python

PERIODIC-II SYLLABUS

Ch-6 Dictionary

Ch-7 Database Concepts

SYLLABUS OF FINAL EXAMS

Ch-1 Computer System

Ch-2 Getting Started with Python

Ch-3 Python Programming Fundamental

Ch-4 Conditional and Looping Constructs

Ch-5 Lists in Python

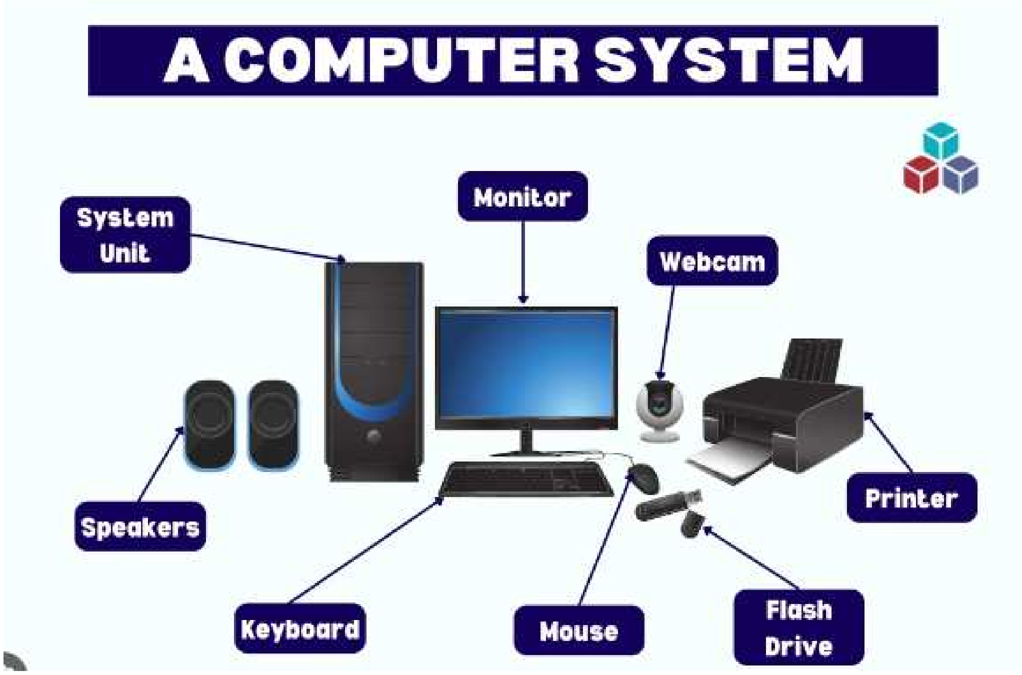
Ch-6 Dictionary

Ch-7 Database Concepts

Ch-8 Structured Query Language

LESSON PLAN

April – Chapter-1 Introduction to Computer System



Topics:

- Introduction to computer and computing: evolution of computing devices, components of a computer system and their interconnections, Input/output devices.
- Computer Memory: Units of memory, types of memory – primary and secondary, data deletion, its recovery and related security concerns.
- Software: purpose and types – system and application software, generic and specific purpose software.

Learning Objectives:

- Understand the evolution and components of computing devices.
- Identify and describe different types of computer memory.
- Explain the purpose and types of software.

Activities/Projects:

- Hands-on activities to identify components of a computer system.
- Demonstration and practical exercises on primary and secondary memory.
- Software installation and usage demonstrations.

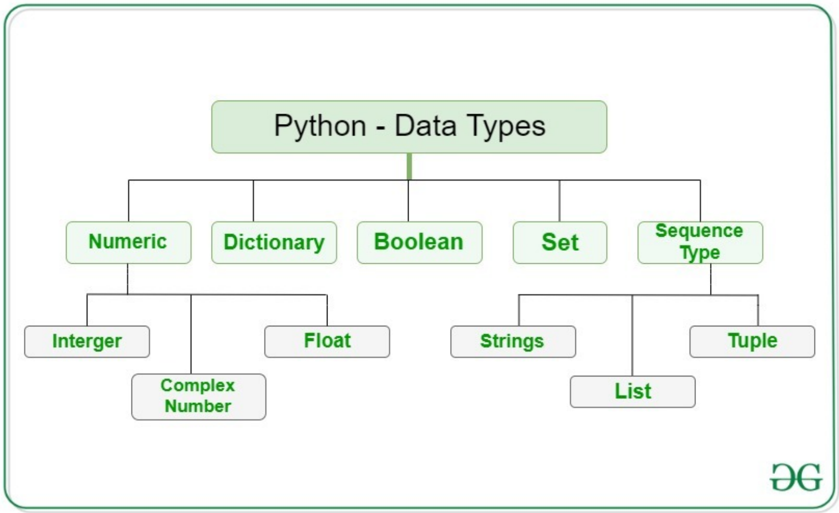
Expected Learning Outcomes:

- Students will grasp the evolution and components of computing devices.
- They will differentiate between primary and secondary memory and understand their implications.
- Students will classify software based on its purpose and functionality.

Assessment:

- Quizzes on computer components and memory types.
- Practical tests on software installation and usage.

May Chapter-2 Basics of Python Programming



Topics:

- Introduction to Python: Features and significance.
- Python Interpreter: Interactive mode vs. Script mode.
- Structure of a Python program: Indentation, statements, and syntax rules.

Learning Objectives:

- Understand the basics and significance of Python programming.
- Differentiate between interactive mode and script mode in Python.
- Learn the fundamental structure and syntax of a Python program.

Activities/Projects:

- Hands-on activities to execute Python commands in interactive mode.
- Writing and running simple Python scripts in script mode.
- Exercises on indentation and structuring Python programs correctly.

Expected Learning Outcomes:

- Students will understand the basic concepts and applications of Python.
- They will be able to use Python in both interactive and script modes.
- Students will learn to write well-structured Python programs.

Assessment:

- Quizzes on Python fundamentals and modes of execution.
- Practical tests on writing and running basic Python programs.

May– Chapter-3 Python Programming Fundamental

Statements
Tokens
Indentation
Fundamentals of Python
constant
Comments
Variables

Topics:

- Indentation: Importance in Python.
- Identifiers & Keywords: Naming conventions and reserved words.
- Constants & Variables: Definition and usage.
- Operators & Precedence: Types of operators and order of execution.
- Data Types: Mutable vs. Immutable.
- Statements & Expressions: Execution and evaluation.
- Comments: Importance and types.
- Input & Output Statements: Handling user interaction.
- Data Type Conversion: Implicit and explicit conversion.
- Debugging: Identifying and fixing errors in Python.

Learning Objectives:

- Understand the basic building blocks of Python programs.
- Identify different operators, their precedence, and data types.
- Differentiate between mutable and immutable data types.
- Learn to write, evaluate, and debug Python programs effectively.

Activities/Projects:

- Hands-on exercises on writing Python code with correct indentation.
- Practical tasks on using variables, operators, and expressions.
- Debugging exercises to identify and fix syntax and logical errors.
- Implementing data type conversion in small Python programs.

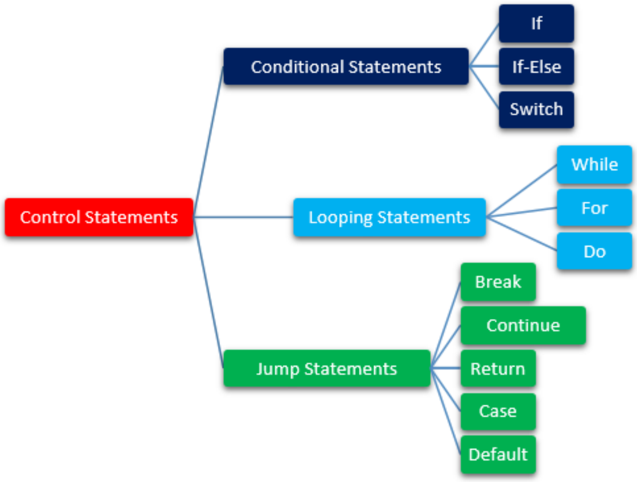
Expected Learning Outcomes:

- Students will be able to write syntactically correct Python programs.
- They will understand how operators and expressions are evaluated.
- Students will effectively use comments, input/output statements, and debugging techniques.

Assessment:

- Quizzes on Python syntax, operators, and data types.
- Practical tests on writing and debugging Python programs.

JULY Ch-4 Conditional and Looping Constructs



Topics:

- Control Statements: Importance and usage.
- Conditional Statements:
  - if-else: Decision-making in Python.
  - if-elif-else: Handling multiple conditions.
- Looping Constructs:
  - while loop: Condition-based iteration.
  - for loop: Iteration over sequences.

Learning Objectives:

- Understand the role of control statements in Python programming.
- Learn how to use conditional statements for decision-making.
- Implement loops for executing repetitive tasks efficiently.

Activities/Projects:

- Hands-on exercises on writing if-else and if-elif-else conditions.
- Practical tasks on using while and for loops in Python.
- Writing small programs to demonstrate different looping constructs.
- Debugging exercises to identify logical errors in loops and conditions.

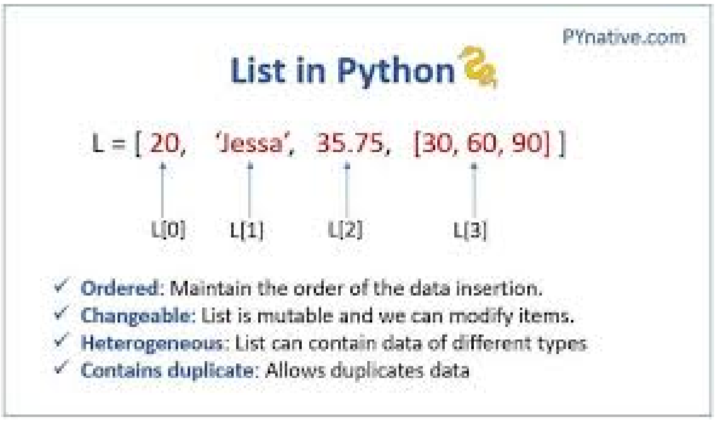
Expected Learning Outcomes:

- Students will be able to write programs using conditional statements.
- They will implement loops for efficient code execution.
- Students will understand and debug common errors in control structures.

Assessment:

- Quizzes on conditional and looping statements.
- Practical tests on implementing if-else, if-elif-else, while, and for loops in Python.

AUGUST :-Chapter:-5 Lists in Python



PYnative.com

List in Python

L = [20, 'Jessa', 35.75, [30, 60, 90]]

✓ Ordered: Maintain the order of the data insertion.
✓ Changeable: List is mutable and we can modify items.
✓ Heterogeneous: List can contain data of different types
✓ Contains duplicate: Allows duplicates data

Topics:

- Introduction to Lists: Definition, characteristics, and uses.
- List Operations:
  - Creating and initializing lists.
  - Traversing and manipulating lists.
- List Methods and Built-in Functions:
  - len(), list(), append(), insert(), count(), index().
  - remove(), pop(), reverse(), sort(), min(), max(), sum().

Learning Objectives:

- Understand the concept and significance of lists in Python.
- Learn how to create, modify, and manipulate lists effectively.
- Explore built-in list methods and their applications.

Activities/Projects:

- Hands-on exercises on creating and modifying lists.
- Practical tasks using various list methods and functions.
- Writing programs to perform list operations like sorting, searching, and reversing.
- Debugging exercises related to list operations.

Expected Learning Outcomes:

- Students will be able to create and manipulate lists efficiently.
- They will understand the usage of built-in list methods and functions.
- Students will be able to apply lists effectively in real-world programming scenarios.

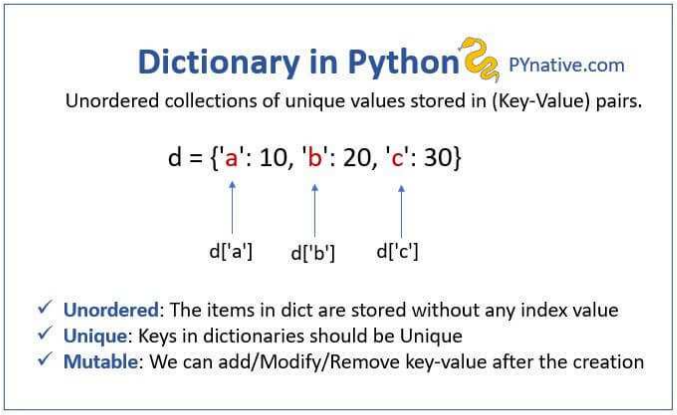
Assessment:

- Quizzes on list operations, methods, and functions.
- Practical tests on implementing list-based programs in Python.

SEPTEMBER

Revision + Half yearly exam

OCTOBER Lesson:-6 Plan: Dictionary in Python



Dictionary in Python PYnative.com

Unordered collections of unique values stored in (Key-Value) pairs.

d = {'a': 10, 'b': 20, 'c': 30}

✓ **Unordered**: The items in dict are stored without any index value
✓ **Unique**: Keys in dictionaries should be Unique
✓ **Mutable**: We can add/Modify/Remove key-value after the creation

Topics:

- Introduction to Dictionary: Concept of key-value pairs and their significance.
- Dictionary Operations:
  - Creating and initializing dictionaries.
  - Traversing, updating, and deleting dictionary elements.
- Dictionary Methods and Built-in Functions:
  - dict(), len(), keys(), values(), items().
  - update(), del(), clear().

Learning Objectives:

- Understand the concept and importance of dictionaries in Python.
- Learn how to create, update, and delete dictionary elements.
- Explore built-in dictionary methods and functions.

Activities/Projects:

- Hands-on exercises on creating and modifying dictionaries.
- Practical tasks using various dictionary methods and functions.
- Writing programs to perform operations like searching, updating, and deleting dictionary elements.
- Debugging exercises related to dictionary operations.

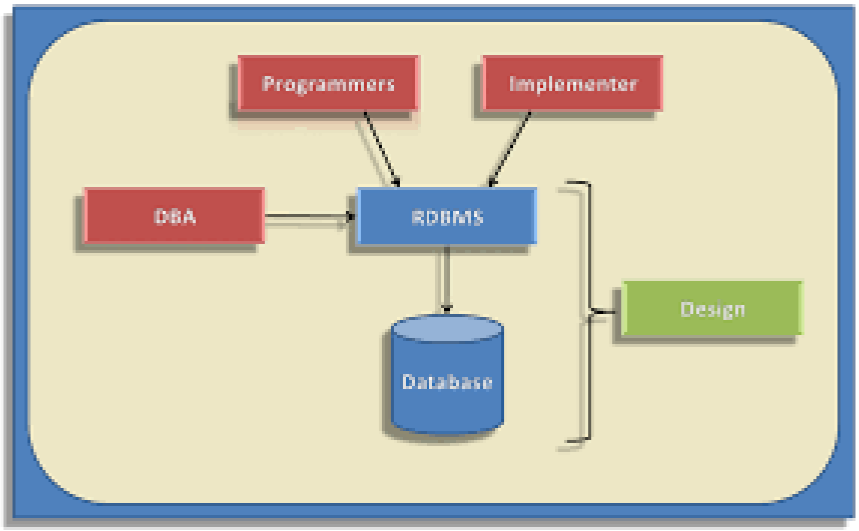
Expected Learning Outcomes:

- Students will be able to create and manipulate dictionaries effectively.
- They will understand the use of key-value pairs in data storage and retrieval.
- Students will be able to apply dictionary functions in real-world programming scenarios.

Assessment:

- Quizzes on dictionary operations, methods, and functions.
- Practical tests on implementing dictionary-based programs in Python.

OCTOBER Lesson 7 : Database Concepts



Topics:

- Introduction to Database Concepts: Definition, need, and importance of databases.
- Database Management System (DBMS): Role and significance.
- Relational Data Model:
  - Key Concepts: Domain, Tuple, Relation.
  - Keys in a Database: Candidate Key, Primary Key, Alternate Key.
- Structured Query Language (SQL):
  - Advantages of SQL over traditional file systems.
  - Types of SQL Commands:
    - Data Definition Language (DDL): Commands for defining structures.
    - Data Query Language (DQL): Commands for retrieving data.
    - Data Manipulation Language (DML): Commands for modifying data.

Learning Objectives:

- Understand the fundamentals and importance of databases.
- Learn key relational data model concepts and different types of keys.
- Explore the advantages of SQL and its different command categories.

Activities/Projects:

- Hands-on activities to explore database concepts using MySQL.
- Identifying primary keys and candidate keys in given datasets.
- Writing SQL commands for data definition, retrieval, and manipulation.
- Group discussions on the advantages of databases over traditional storage systems.

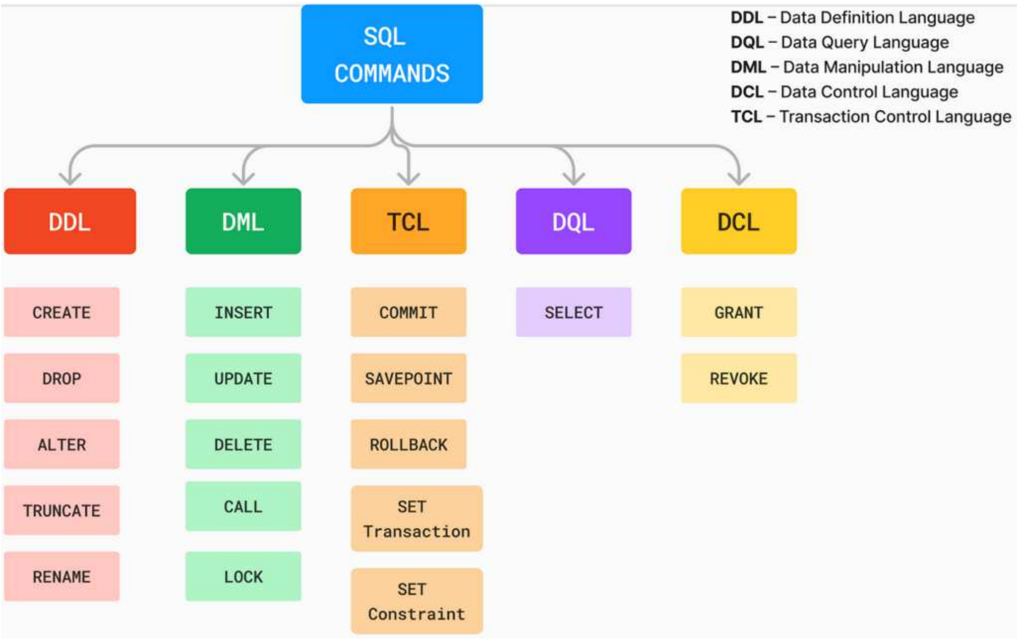
Expected Learning Outcomes:

- Students will understand the need for databases and DBMS.
- They will be able to identify different keys in a relational database.
- Students will learn the different SQL command categories and their applications.

Assessment:

- Quizzes on database concepts, relational models, and SQL.
- Practical tests on identifying keys and executing SQL commands in MySQL.

November Lesson 8: Structured Query Language (SQL) with MySQL



Topics:

- Introduction to MySQL: Overview and significance of SQL.
- Creating a Database in MySQL: Understanding databases and data types.
- Data Definition (DDL Commands):
  - CREATE DATABASE, CREATE TABLE, DROP, ALTER.
- Data Query (DQL Commands):
  - SELECT, FROM, WHERE with relational operators.
  - BETWEEN, logical operators, IS NULL, IS NOT NULL.
- Data Manipulation (DML Commands):
  - INSERT, DELETE, UPDATE.

Learning Objectives:

- Understand the basics of SQL and MySQL.
- Learn how to create and manage databases and tables.
- Explore querying data using different SQL commands.
- Perform data manipulation operations in MySQL.

Activities/Projects:

- Hands-on exercises on creating databases and tables.
- Writing SQL queries using SELECT, WHERE, and logical operators.
- Performing INSERT, DELETE, and UPDATE operations on data.
- Debugging and fixing SQL query errors.

Expected Learning Outcomes:

- Students will be able to create and manage databases in MySQL.
- They will write SQL queries for retrieving and manipulating data.
- Students will understand relational operators and logical conditions in SQL.

Assessment:

- Quizzes on SQL commands and MySQL concepts.
- Practical tests on database creation, data retrieval, and manipulation using SQL.

DECEMBER /JANUARY

Final Practical & Revision of Final Examination

FEBRUARY/MARCH

Final Exam