

CLASS-7TH
LESSON PLAN
2026-27

MONTH-APRIL

Lesson – Computer Arithmetic

Topics:

- Introduction to Number System
- Decimal Number System
- Binary Number System
- Octal Number System
- Hexadecimal Number System
- Conversion between Number Systems
- Binary Arithmetic (Addition and Subtraction)

Learning Objectives:

- Understand the concept of number systems used in computers.
- Identify the Decimal, Binary, Octal, and Hexadecimal number systems.
- Differentiate between various number systems based on their bases.
- Learn the methods of converting numbers from one number system to another.
- Perform simple binary addition and subtraction.
- Develop logical and computational thinking through number system operations.

Art & Integrated Activity / Project / Practical:

- Prepare a chart showing the four number systems and their bases.
- Create a comparison table of Decimal, Binary, Octal, and Hexadecimal number systems.
- Practical activity: Convert given decimal numbers into binary, octal, and hexadecimal forms.
- Solve binary addition and subtraction problems in groups.
- Role-play activity where students represent binary digits (0 and 1) to demonstrate binary counting.
- Project work on "**How Computers Understand Numbers**" with diagrams and examples.
- Design a colourful number system wheel showing the relationship between different number systems.

Expected Learning Outcomes:

Students will:

- Understand the importance of number systems in computer operations.
- Identify the bases and symbols used in different number systems.
- Convert numbers between Decimal, Binary, Octal, and Hexadecimal systems.
- Perform simple binary arithmetic accurately.
- Apply logical reasoning to solve number system problems.
- Develop confidence in using computer arithmetic concepts in practical activities.

Assignment and Assessments/Test:

- Assignments on different number systems and their bases.
- Worksheets on Decimal, Binary, Octal, and Hexadecimal conversions.
- Practical exercises on number system conversion and binary arithmetic.
- Quizzes on computer arithmetic concepts and binary operations.
- Activity-based assessment through chart preparation, classroom discussions, and project presentations.

Remedial Measures:

- Extra practice sessions on number system conversions using simplified examples.
- Demonstration videos explaining binary, octal, decimal, and hexadecimal number systems.
- Step-by-step classroom practice for binary addition and subtraction.
- One-to-one guidance and additional worksheets for students requiring extra support.
- Revision quizzes, flash cards, and interactive activities to strengthen understanding of computer arithmetic concepts.

MONTH-MAY

Lesson – AI (Techniques of AI and Sustainable Development)

Topics:

- Introduction to AI Techniques
- Data Collection and Data Analysis
- Machine Learning Basics
- Problem Solving Using AI
- Introduction to Sustainable Development
- AI for Sustainable Development Goals (SDGs)
- Real-life Applications of AI for Environmental and Social Development

Learning Objectives:

- Understand the basic techniques used in Artificial Intelligence.
- Learn how AI collects, processes, and analyzes data to make decisions.
- Identify the role of Machine Learning in solving real-world problems.
- Understand the concept of Sustainable Development and the Sustainable Development Goals (SDGs).

- Explore how AI contributes to environmental protection, healthcare, education, agriculture, and smart cities.
- Develop awareness of responsible and ethical use of AI for a sustainable future.

Art & Integrated Activity / Project / Practical:

- Prepare a poster illustrating AI techniques and their real-life applications.
- Create a chart showing how AI supports different Sustainable Development Goals (SDGs).
- Group discussion on the role of AI in protecting the environment.
- Practical activity: Identify AI-based applications that help reduce pollution, save energy, or improve healthcare.
- Role-play activity demonstrating AI solutions for smart farming, waste management, or disaster prediction.
- Project work on "**AI for a Better and Sustainable Future**" using real-life examples.
- Design an infographic showing how AI can improve education, healthcare, agriculture, and environmental conservation.

Expected Learning Outcomes:

Students will:

- Understand the basic techniques used in Artificial Intelligence.
- Explain how AI uses data to make intelligent decisions.
- Identify the importance of Sustainable Development and the SDGs.
- Describe how AI contributes to solving environmental and social challenges.
- Develop awareness of ethical and responsible use of AI.
- Apply AI concepts through classroom discussions, activities, and project work.

Assignment and Assessments/Test:

- Assignments on AI techniques and Sustainable Development.
- Worksheets based on data analysis, Machine Learning, and SDGs.
- Quizzes on AI concepts, techniques, and real-life applications.
- Activity-based assessments through posters, charts, and presentations.
- Project evaluation on AI solutions for sustainable development and community welfare.

Remedial Measures:

- Extra practice sessions using simplified notes on AI techniques and Sustainable Development.
- Video demonstrations showing AI applications in healthcare, agriculture, education, and environmental protection.
- Classroom discussions using real-life examples of AI supporting sustainable living.
- One-to-one guidance and additional worksheets for students requiring extra support.
- Revision quizzes, interactive classroom activities, and project reviews to strengthen understanding of AI techniques and Sustainable Development.

MONTH –JULY

Lesson – Formulas and Functions in MS Excel

Topics:

- Introduction to Formulas in MS Excel
- Components of a Formula
- Cell References (Relative, Absolute, and Mixed – Introduction)
- Using Arithmetic Operators (+, –, *, /)
- Introduction to Functions
- Common Functions: **SUM()**, **AVERAGE()**, **MAX()**, **MIN()**, **COUNT()**
- Applying Formulas and Functions in Real-life Worksheets

Learning Objectives:

- Understand the concept and importance of formulas in MS Excel.
- Learn how to create and edit simple formulas using arithmetic operators.
- Identify different types of cell references.
- Understand the purpose and use of built-in Excel functions.
- Apply functions such as **SUM()**, **AVERAGE()**, **MAX()**, **MIN()**, and **COUNT()** to solve real-life problems.
- Develop accuracy and logical thinking while performing calculations in spreadsheets.

Art & Integrated Activity / Project / Practical:

- Prepare a colourful marks sheet using MS Excel and calculate the total using the **SUM()** function.
- Create a class attendance sheet and use the **COUNT()** function to count entries.
- Design a monthly expenses worksheet and calculate the average expenditure using the **AVERAGE()** function.
- Practical activity: Enter data and apply formulas to perform addition, subtraction, multiplication, and division.
- Group activity: Compare students' marks using **MAX()** and **MIN()** functions.
- Project work on "**Using Excel Formulas in Daily Life**" with examples from school, home, or business.
- Create a sales report and use different functions to analyze the data.

Expected Learning Outcomes:

Students will:

- Understand the difference between formulas and functions.
- Create simple formulas using arithmetic operators.
- Apply basic Excel functions accurately.
- Use formulas and functions to perform calculations in worksheets.
- Analyze and organize data efficiently using Excel.
- Develop confidence in solving practical problems using spreadsheets.

Assignment and Assessments/Test:

- Assignments on Excel formulas and functions.
- Worksheets based on arithmetic formulas and cell references.
- Practical exercises using **SUM()**, **AVERAGE()**, **MAX()**, **MIN()**, and **COUNT()** functions.
- Quizzes on Excel formulas, functions, and spreadsheet concepts.
- Activity-based assessment through worksheet creation and project presentation.

Remedial Measures:

- Extra practice sessions using simplified examples of formulas and functions.
- Demonstration videos explaining the use of Excel formulas and built-in functions.
- One-to-one guidance during practical sessions for students requiring additional support.
- Additional worksheets for practicing formulas, functions, and cell references.
- Revision quizzes and hands-on classroom activities to strengthen understanding of Excel calculations and data analysis.

MONTH-AUGUST

Lesson–AI (AI in Industries)

Topics:

- Introduction to Artificial Intelligence in Industries
- AI in Healthcare
- AI in Education
- AI in Transport
- AI in Communication
- Advantages of AI in Different Industries
- Real-life Case Studies and Applications of AI

Learning Objectives:

- Understand the role of Artificial Intelligence in various industries.
- Identify the applications of AI in healthcare, education, transport, and communication.
- Learn how AI improves efficiency, accuracy, safety, and decision-making.
- Recognize the advantages of AI in solving real-world problems.
- Develop awareness of AI technologies used in daily life and different professions.
- Build critical thinking by analyzing real-life AI case studies.

Art & Integrated Activity / Project / Practical:

- Prepare posters illustrating AI applications in different industries.
- Create charts comparing traditional methods with AI-based solutions.
- Group discussion on the benefits and challenges of AI in everyday life.

- Role-play activities demonstrating AI use in hospitals, schools, transport, and communication.
- Case study presentation on AI applications in healthcare, education, or transport.
- Practical activity: Identify AI-enabled applications used at home or school and explain their functions.
- Project work on "How AI is Transforming Industries" with real-life examples.

Expected Learning Outcomes:

Students will:

- Understand the importance of AI in modern industries.
- Identify various AI applications in healthcare, education, transport, and communication.
- Explain how AI improves productivity, efficiency, and decision-making.
- Appreciate the benefits of AI in solving real-world challenges.
- Develop awareness of responsible and effective use of AI technologies.
- Apply AI concepts to everyday situations through discussions and project work.

Assignment and Assessments/Test:

- Assignments on AI applications in different industries.
- Worksheets based on AI in healthcare, education, transport, and communication.
- Quizzes on AI concepts, industrial applications, and advantages.
- Activity-based assessments through case studies and presentations.
- Project evaluation on real-life AI applications and their impact on society.

Remedial Measures:

- Extra practice sessions using simplified notes on AI applications.
- Video demonstrations showing AI in hospitals, schools, transport, and communication.
- Classroom discussions using real-life examples to reinforce concepts.
- One-to-one guidance and additional worksheets for students requiring extra support.
- Revision quizzes and interactive activities to strengthen understanding of AI applications.

Month-October

Lesson – HTML and CSS

Topics:

- Introduction to HTML
- Basic Structure of an HTML Document
- HTML Tags
- Common HTML Tags and Their Uses
- Creating Headings and Paragraphs
- Creating Lists in HTML

- Creating Tables in HTML
- Introduction to CSS
- Importance and Uses of CSS
- Styling Lists and Tables using CSS
- Basic CSS Properties

Learning Objectives:

- Understand the meaning and purpose of HTML.
- Learn the basic structure of an HTML document.
- Identify and use common HTML tags.
- Learn how to create headings, paragraphs, lists, and tables.
- Understand the meaning and importance of CSS.
- Learn how CSS is used to style HTML elements.
- Apply basic CSS properties to lists and tables.
- Develop basic web page designing and formatting skills.

Art & Integrated Activity / Project / Practical:

- Prepare a chart showing common HTML tags and their functions.
- Create a simple HTML webpage containing headings, paragraphs, and lists.
- Practical activity: Create ordered and unordered lists using HTML tags.
- Practical activity: Create a table displaying student information such as name, class, and subject.
- Create a comparison chart showing HTML and CSS and their purposes.
- Demonstrate how CSS changes the appearance of an HTML webpage.
- Practical activity: Apply CSS to change the font, size, alignment, and border of lists and tables.
- Design a simple "My School" webpage using HTML and CSS.
- Project work on "**Design Your Own Webpage**" using HTML tags and basic CSS styling.

Expected Learning Outcomes:

Students will:

- Understand the purpose and importance of HTML and CSS.
- Identify and use common HTML tags correctly.
- Create basic webpages using HTML.
- Create and format lists and tables using HTML.
- Understand the role of CSS in webpage designing.
- Apply basic CSS properties to HTML elements.
- Style lists and tables using CSS.
- Develop basic skills in creating attractive and well-structured webpages.

Assignment and Assessments/Test:

- Assignments based on HTML tags and their uses.
- Worksheets identifying HTML tags and their functions.
- Practical exercises on creating lists and tables.

- Quiz on HTML, CSS, and basic web designing concepts.
- Activity-based assessment through webpage designing.
- Practical assessment based on HTML and CSS coding.
- Project evaluation of the "**Design Your Own Webpage**" activity.

Remedial Measures:

- Extra practice sessions using simplified notes on HTML and CSS.
- Step-by-step demonstrations of creating webpages.
- Video demonstrations explaining HTML tags and CSS styling.
- Additional worksheets on HTML tags, lists, and tables.
- Practice exercises on applying basic CSS properties.
- One-to-one guidance for students requiring additional support.
- Pair activities to help students identify and correct coding errors.
- Revision quizzes and interactive activities to strengthen HTML and CSS concepts.

Mont-November

Lesson – Control Structures in Python

Topics:

- Operators in Python
- Types of Operators
- Arithmetic Operators
- Relational and Logical Operators
- Assignment Operators
- Operator Precedence
- Rules of Operator Precedence
- Control Structures in Python
- Conditional Statements
- `if` Statement
- `if-else` Statement
- `if-elif-else` Statement
- Simple Programs using Operators and Conditionals

Learning Objectives:

- Understand the meaning and purpose of operators in Python.
- Identify and use different types of Python operators.
- Understand the concept of operator precedence.
- Learn how Python evaluates expressions using operator precedence.
- Understand the need for control structures in programming.
- Learn how conditional statements control the flow of a program.
- Use `if`, `if-else`, and `if-elif-else` statements in Python programs.
- Develop logical thinking and decision-making skills through programming.

Art & Integrated Activity / Project / Practical:

- Prepare a chart showing different types of Python operators with examples.

- Create a table explaining the order of operator precedence.
- Group activity to solve expressions using different operators and determine their results.
- Create a flowchart showing the working of an `if-else` statement.
- Practical activity: Write programs to perform arithmetic calculations using operators.
- Practical activity: Write programs to check whether a number is positive, negative, or zero.
- Practical activity: Write a program to find the greater of two or three numbers using conditional statements.
- Create simple real-life decision-making examples using `if-else`.
- Project work on "**Python Decision Making in Real Life**" using flowcharts and simple programs.

Expected Learning Outcomes:

Students will:

- Understand different types of operators used in Python.
- Apply operators correctly in Python expressions.
- Explain and use operator precedence.
- Understand the purpose of control structures.
- Differentiate between different conditional statements.
- Write Python programs using `if`, `if-else`, and `if-elif-else`.
- Use logical thinking to solve simple programming problems.
- Develop basic decision-making and problem-solving skills.

Assignment and Assessments/Test:

- Assignments based on Python operators and their uses.
- Worksheets on evaluating expressions using operator precedence.
- Quizzes on operators and conditional statements.
- Flowchart-based activities for different decision-making situations.
- Practical assessment based on writing programs using operators and conditionals.
- Class test on control structures and Python expressions.
- Project evaluation based on "**Python Decision Making in Real Life**."

Remedial Measures:

- Extra practice sessions using simple examples of Python operators.
- Step-by-step demonstrations of operator precedence and expression evaluation.
- Additional worksheets on conditional statements.
- Video demonstrations explaining the use of `if`, `if-else`, and `if-elif-else`.
- Practice activities using simple real-life decision-making situations.
- One-to-one guidance for students requiring additional support.
- Pair programming activities to identify and correct errors in conditional statements.
- Revision quizzes and interactive activities to strengthen understanding of operators and control structures.

Month-December

Lesson – Data Visualisation and Analysis

Topics:

- What is Data?
- Structured and Unstructured Data
- Simple Methods of Data Collection
- Understanding Digital Data

Learning Objectives:

- Understand the meaning and importance of data in everyday life and computing.
- Identify different types of data and distinguish between structured and unstructured data.
- Understand simple methods used to collect data.
- Recognise examples of digital data used in computers and digital devices.
- Differentiate between structured and unstructured data using real-life examples.
- Develop basic skills in collecting, organising, and interpreting data.

Art & Integrated Activity / Project / Practical:

- Prepare a colourful chart explaining "**What is Data?**" with real-life examples.
- Create a comparison table showing the differences between structured and unstructured data.
- Classroom activity: Collect simple data such as students' favourite subjects, colours, or hobbies.
- Organise the collected data into a simple table and discuss the observations.
- Activity on digital data: Identify examples of text, images, audio, video, and numbers as forms of digital data.
- Group project on "**Data Around Us**" showing examples of data collected from school and daily life.
- Create a simple poster or infographic showing different methods of data collection.

Expected Learning Outcomes:

Students will:

- Explain the meaning and importance of data.
- Identify different forms of data in everyday situations.
- Differentiate between structured and unstructured data.
- Identify simple methods of collecting data.
- Recognise different types of digital data such as text, images, audio, video, and numbers.
- Organise simple collected data in a meaningful way.
- Develop logical thinking and basic data-handling skills.

Assignment and Assessments/Test:

- Assignments on the meaning and types of data.
- Worksheets on structured and unstructured data.
- Activities based on identifying different forms of digital data.
- Practical exercises involving simple data collection and organisation.
- Short quizzes on data, data collection methods, and digital data.
- Activity-based assessment through charts, group activities, classroom discussions, and project presentations.

Remedial Measures:

- Provide simple real-life examples to explain the concept of data.
- Use pictures, charts, and classroom objects to demonstrate different types of data.
- Conduct step-by-step activities for distinguishing structured and unstructured data.
- Provide additional worksheets with simple data collection exercises.
- Give one-to-one guidance to students who need additional support.
- Use revision activities, flash cards, and interactive quizzes to reinforce key concepts.

Month – January

Lesson – 7: Loops in Python

Topics:

- Introduction to Loops
- Types of Loops
- Introduction to Turtle
- Programming with Turtle
- Drawing with Turtle

Learning Objectives:

- Understand the concept and importance of loops in Python programming.
- Identify and differentiate between different types of loops.
- Understand how loops help in repeating a set of instructions.
- Learn the basic concepts of the Python Turtle module.
- Use Turtle commands to control the movement of the turtle.
- Create simple drawings and patterns using Turtle programming.
- Develop logical thinking and problem-solving skills through programming activities.

Art & Integrated Activity / Project / Practical:

- Prepare a flowchart explaining how a loop works in a Python program.
- Create a comparison chart of different types of loops with simple examples.
- Practical activity: Write simple Python programs using `for` and `while` loops.
- Introduction activity with Turtle to understand commands such as movement, turning, pen up, and pen down.
- Draw basic shapes such as a square, triangle, rectangle, and polygon using Turtle.
- Create colourful patterns and repeated designs using loops with Turtle.

- Project work on "**Creative Drawing with Python Turtle**" by designing a simple digital artwork using loops.

Expected Learning Outcomes:

Students will:

- Explain the purpose of loops in Python.
- Identify and differentiate between different types of loops.
- Use loops to repeat instructions efficiently.
- Understand the basic working of the Turtle module.
- Write simple Turtle programs using appropriate commands.
- Draw basic shapes and patterns using Turtle.
- Combine loops and Turtle commands to create creative designs.
- Develop logical, computational, and problem-solving skills through practical programming.

Assignment and Assessments/Test:

- Assignments based on loops and their types.
- Worksheets on identifying suitable loops for given situations.
- Practical exercises using `for` and `while` loops.
- Turtle programming exercises for drawing basic shapes.
- Practical test involving loops and Turtle commands.
- Quiz on loops, Turtle commands, and basic programming concepts.
- Activity-based assessment through Turtle drawings, classroom discussions, and project presentations.

Remedial Measures:

- Explain loops using simple real-life examples such as repeated daily activities.
- Provide step-by-step demonstrations of `for` and `while` loops.
- Use simple programs before introducing more complex loop-based activities.
- Demonstrate Turtle commands through guided practical exercises.
- Provide additional worksheets and practice programs for students requiring support.
- Give one-to-one guidance during programming activities.
- Use revision exercises and simple Turtle drawing tasks to strengthen understanding of loops and programming concepts.