



S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

1. Basic Model of Computation

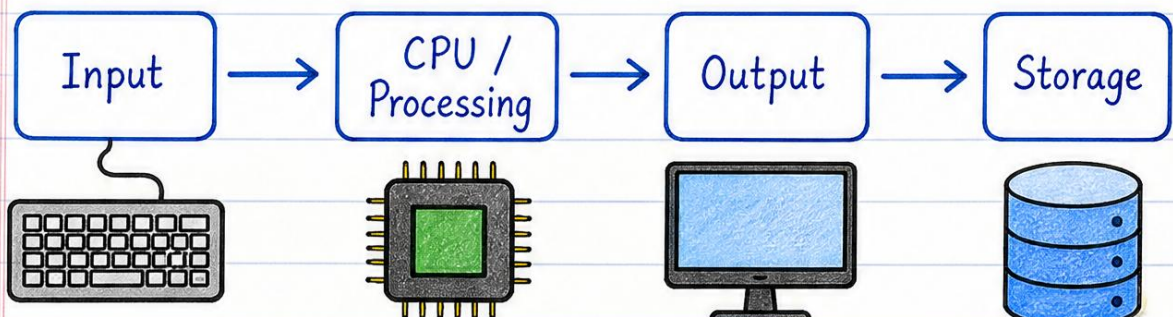
Computer एक electronic machine है जो data को input के रूप में स्वीकार करके instructions के अनुसार process करता है और meaningful result अर्थात output देता है।

Computation Model के मुख्य भाग:

1. **Input:** User या अन्य device से data और instructions प्राप्त करना।
2. **Processing:** CPU instructions को execute करके data पर operations करता है।
3. **Output:** Processed information को screen, printer या file के रूप में देता।
4. **Storage:** Data और results को भविष्य में उपयोग के लिए सुरक्षित रखना।
5. **Control:** सभी activities को उचित क्रम में manage करना।

Working cycle: Input → Processing → Output → Storage

उदाहरण: Keyboard से 2 और 3 input देने पर CPU addition करता है और screen पर 5 output दिखाता है।





S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

2. Algorithms



Algorithm किसी problem को solve करने के लिए लिखे गए स्पष्ट, क्रमबद्ध और finite steps का समूह है।

एक अच्छे Algorithm की विशेषताएँ:

1. **Input:** आवश्यक data स्पष्ट रूप से defined होना चाहिए।
2. **Output:** कम-से-कम एक निश्चित result प्राप्त होना चाहिए।
3. **Definiteness:** प्रत्येक step का अर्थ clear और unambiguous होना चाहिए।
4. **Finiteness:** Algorithm सीमित steps के बाद समाप्त होना चाहिए।
5. **Effectiveness:** प्रत्येक step practical और executable होना चाहिए।

Algorithm लिखने का सामान्य तरीका:

Problem को समझें → Input और Output पहचानें →
Steps लिखें → Dry run करें → Result जाँचें।

उदाहरण: दो numbers में बड़ा number ज्ञात करना।

- Step 1: दो numbers A और B input करें।
- Step 2: यदि $A > B$ है, तो A बड़ा है।
- Step 3: अन्यथा B बड़ा है।
- Step 4: बड़ा number print करें।





S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

3. Flowcharts

Flowchart किसी Algorithm या process का graphical representation है। इसमें standard symbols और arrows की सहायता से steps दिखाए जाते हैं।

मुख्य Flowchart Symbols:

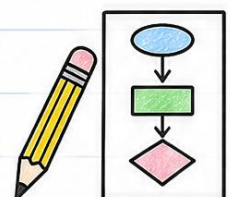
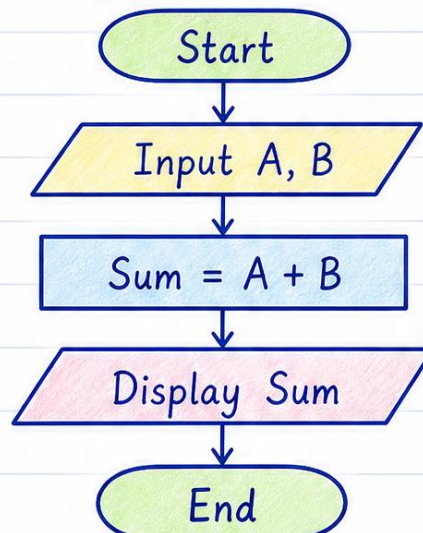
- Oval: Start या End को दर्शाता है।
- Parallelogram: Input या Output के लिए।
- Rectangle: Processing या calculation के लिए।
- Diamond: Decision या condition के लिए।
- Arrow: Control flow की दिशा दिखाता है।

Flowchart के लाभ:

Problem को आसानी से समझना, logic की गलतियाँ पहचानना, communication बेहतर करना और program लिखना सरल बनाना।

उदाहरण flow:

Start → दो numbers Input → $Sum = A + B$ →
Sum Display → End





S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

4. Programming Languages

Programming Language वह भाषा है जिसका उपयोग programmer computer को instructions देने के लिए करता है।

Programming Languages के प्रकार:

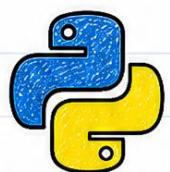
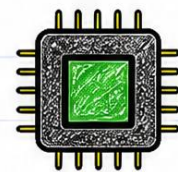
1. **Machine Language:** Binary digits 0 और 1 में लिखी जाती है। यह computer द्वारा सीधे समझी जाती है, परन्तु मानव के लिए कठिन होती है।
2. **Assembly Language:** Mnemonic codes जैसे ADD, SUB और MOV का उपयोग करती है। इसे Assembler translate करता है।
3. **High-Level Language:** मानव के लिए सरल और readable होती है। उदाहरण: Python, C, C++, Java।

Python एक High-Level, interpreted और general-purpose Programming Language है।

Translator का काम source code को machine-understandable form में बदलना है।



01010101
10110011
11001010





S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

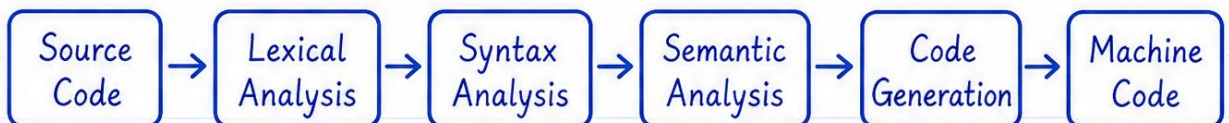
5. Compilation

Compilation वह process है जिसमें High-Level Language में लिखे गए source code को Translator machine code या intermediate code में बदलता है।

Compiler पूरे source program को एक साथ translate करता है और errors की list देता है।

Interpreter program को line-by-line translate और execute करता है।

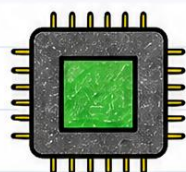
Compiler के मुख्य चरण:



Compiler के लाभ: execution तेज हो सकता है, errors compilation के समय मिलते हैं और executable file बनाई जा सकती है।

Interpreter के लाभ: debugging आसान होती है और program को तुरंत line-by-line चलाया जा सकता है।

Compiler	Interpreter
• पूरे source program को एक साथ translate करता है। • errors की list देता है। • execution तेज होती है। • executable file बनाई जा सकती है।	• program को line-by-line translate और execute करता है। • errors तुरंत मिलते हैं। • execution अपेक्षाकृत धीमी होती है। • executable file नहीं बनती है।



Compiler



Interpreter



S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

6. Testing & Debugging

Testing वह process है जिसमें program को अलग-अलग inputs देकर यह जाँचा जाता है कि expected output मिल रहा है या नहीं।

Debugging program में उपस्थित errors को ढूँढ़ने और ठीक करने की प्रक्रिया है।

Errors के प्रकार:

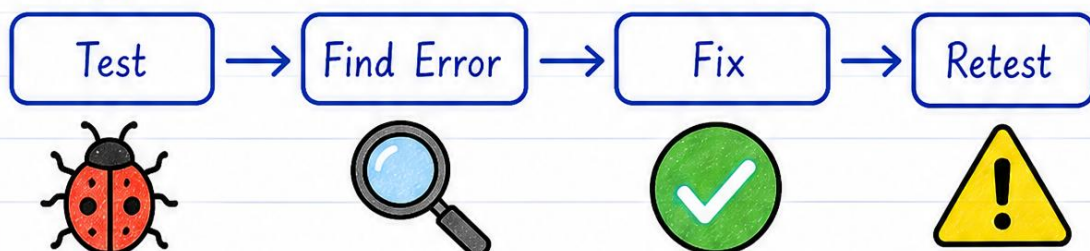
1. **Syntax Error:** Language के rules का उल्लंघन, जैसे bracket या colon की गलती।
2. **Runtime Error:** Program run करते समय होने वाली गलती, जैसे zero से division।
3. **Logical Error:** Program चलता है, परन्तु output गलत आता है।

Testing के चरण:

Test cases बनाना → Program run करना → Actual और Expected output compare करना → Error ठीक करना → फिर से test करना।

Debugging tips:

error message पढ़ें, छोटे भागों में जाँचें, print statements या debugger का उपयोग करें और boundary cases test करें।





S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 1: Introduction to Programming

7. Documentation

Documentation program, उसके उपयोग, design और working के बारे में लिखी गई जानकारी है। यह user और developer दोनों के लिए उपयोगी होती है।

Documentation के प्रकार:

1. **Internal Documentation:** Source code के अंदर comments और meaningful names के रूप में लिखी जाती है।
2. **External Documentation:** Program के बाहर अलग document, user manual या README file के रूप में होती है।

अच्छी Documentation में problem का उद्देश्य, required input, expected output, algorithm, installation steps, usage और limitations स्पष्ट होने चाहिए।

Comments code की logic समझाने के लिए लिखे जाते हैं, लेकिन comments को सही और updated रखना आवश्यक है।

Documentation के लाभ:

Program को समझना आसान होता है, maintenance और debugging सरल होती है, team में communication बेहतर होता है और future changes सुरक्षित रूप से किए जा सकते हैं।

Documentation checklist:

Purpose → Requirements → Algorithm → Usage → Examples →
Limitations → Version information

