



## SWE1301: Introduction to Problem Solving and Software Development

Lecture 04 : Introduction to Programming Languages

Venue : CIT Theatre

Presented by: M. I. Mukhtar



## Lecture Outline

- Introduction to Programming Languages
- Level of Programming Languages
- Language Translators
- Python and its syntax

12/22/2021

SWE1301



## Programming Languages

- Human Languages are known as natural languages.
  - computer cannot understand natural language.
- A programming language is a set of rules that provide a way of telling the computer what operations to perform.
- A person who writes computer programs is called a **Programmer**.
- Different programming languages are designed for different types of programs.

SWE1301\_Introduction to Problem Solving and Software Development

3



## Types/Levels of Programming Languages

- High Level Programming Languages:
- Low Level Programming Languages
- Machine Language



## High Level Language

- A high level language is a programming language such as Java or python that enables a programmer to write programs which is understandable by the programmer(Human) and can perform any task.
- The high level languages are much closer to human language.
- Example of high level language.  

```
Class Triangle{
Float surface()
Return b*h/2;
```



## Low level Language

- The low level language is a language that deals with hardware registers.
- Assembly language is the best example of low level language, this is in between machine language and high level language.
- Programmers still use assembly language when speed is essential or when they need to perform an operation that isn't possible in a high level language.
  - Example of low level language code
  - (Assembly language)  
 Load R1, Load R2



## Machine Language

- Machine language is a system of instructions and data executed directly by the computer's CPU.
- It is the lowest level programming language and the only language that is understood by the computer.
- It is the computer language that is directly executable by the computer without the need of translation by a compiler or an assembler.
- Example:  

```
000011101010001
000100100100001
```

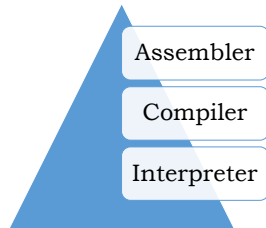


## Language Translator

- The computer only understands object code(machine code)
  - It does not understand any source code(Programming language).
  - There must be a program that converts source code into the object code so that the computer can understand it.
- The Language translator is the one which does that.
  - **Language translator** is a program that converts the source code into the object code.
  - The programmer writes the source code and then translator converts it into machine readable format.

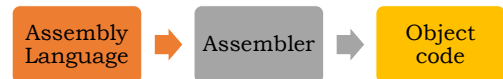


## Types of Language Translators



### Assembler

- Assembler is the language translator that converts assembly language code into the object code.



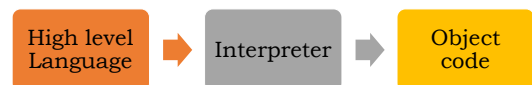
### Compiler

- Compiler is the language translator that converts high level language code into the object code.
- It converts the whole code at a time.



### Interpreter

- Interpreter is a language translator that converts high level language code into the object code.
- It converts the code line by line.





## Python

- Python is an easy to learn, powerful programming language.
- It has efficient **high-level** data structures and a simple but effective approach to **object-oriented programming**.
- Python's elegant syntax and dynamic typing, together with its **interpreted nature**, make it an ideal language for scripting and rapid application development in many areas on most platforms.
- Python uses Interpreter.

SWE1301\_Introduction to Problem Solving and Software Development

13



## Python Data Types

- Some of the data types supported in python include :
  - int
  - float
  - str

SWE1301\_Introduction to Problem Solving and Software Development

14



## Python Input & Output Syntax

- Python uses:
  - `input()`: for input statements
  - `print()`: for output statements

SWE1301\_Introduction to Problem Solving and Software Development

15



## Recall-Structures of Algorithms

- The **sequential structure** executes instructions one after another in a sequence.
- The **decision structure** branches to execute one of two possible sets of instructions.
- The **loop structure** executes a set of instructions many times.

12/22/2021

SWE1301: Problem Solving and Software Development MIM

16



## Sequential Logic Structure

- The most commonly used and the simplest logic structure is the sequential structure.
- All problems **use the sequential structure**, and most problems use it in conjunction with **one or more of the other logic structures**.
- The **sequential structure** executes instructions one after another in a sequence.
- A programmer who uses the sequential logic structure is asking the computer to process a set of instructions in sequence **from the top to the bottom** of an algorithm.

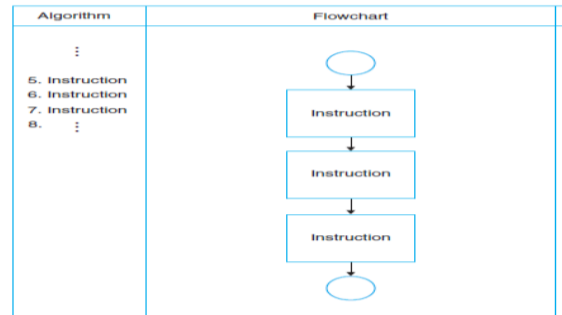
12/22/2021

SWE1301: Problem Solving and Software Development MIM

17



## Sequential Logic Structure Format



12/22/2021

SWE1301: Problem Solving and Software Development MIM

18



## Example 1

- Using appropriate solution tools, solve the problem of calculating the area of a circle.

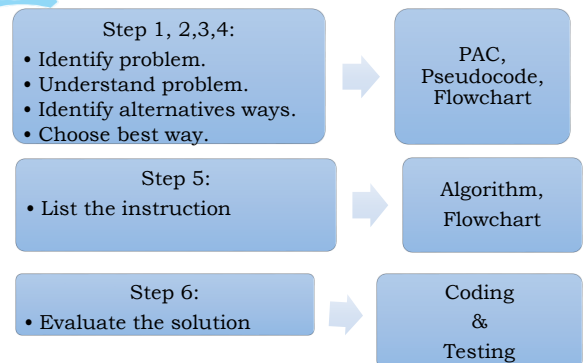
12/22/2021

SWE1301: Problem Solving and Software Development MIM

19



## Recall



12/22/2021

SWE1301: Problem Solving and Software Development MIM

20



### Solution: Problem Analysis Chart..

- Calculate the Area of a circle given the radius (r)
- The Area is calculated by multiplying the  $\pi$  with radius.

Given Data	Required Results
r $\pi$	Area
Processing Required	Solution alternative
Area = $\pi r^2$	1. Define $r^2$ as $r * r$ or $\text{pow}(r,2)$ 2. Define $\pi$ as a constant (assign value to $\pi$ ) or input value.

Programming languages use  $\pi$  instead of  $\pi$  symbol

12/22/2021

SWE1301: Problem Solving and Software Development MIM

21



### Solution: Pseudocode

From Example 1;

- Enter the radius and  $\pi$
- Calculate Area
- Display Area

12/22/2021

SWE1301: Problem Solving and Software Development MIM

22



### Solution: Algorithms...

From Example 1:

- Step 1: Enter radius
- Step 2: Enter  $\pi$
- Step 3: Area =  $\pi * \text{radius} * \text{radius}$
- Step 4: Print Area
- Step 5 : End

Or

- Step 1: Enter radius
- Step 2: Enter  $\pi$
- Step 3: Area =  $\pi * \text{pow}(\text{radius},2)$
- Step 4: Print Area
- Step 5 : End

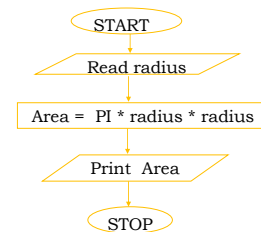
12/22/2021

SWE1301: Problem Solving and Software Development MIM

23



### Solution: Flowchart



12/22/2021

SWE1301: Problem Solving and Software Development MIM

24



## Solution: Coding in Python

### From Example 1

#### 1. Alternative 1

```
import math
radius = float(input("Enter the radius"))
Area = math.pi * radius * radius
print(Area)
```

#### 2. Alternative 2

```
radius = 12
Area = math.pi * pow(radius,2)
print("The area of a circle with radius", r,
"is",Area)
```

12/22/2021

SWE1303

25



## Solution: Testing code for Bugs

- Test with the following
  - Radius of 10
  - Radius of -2

12/22/2021

SWE1301: Problem Solving and Software Development MIM

26



Questions??

27