



A Gentle Introduction to Python (CCC634)

Sumit Shekhar

Assistant Professor

Department of Computer Science and Engineering

Email: sumit.shekhar@snu.edu.in

Evaluation scheme along with Learning Outcome

- Understand the basic concepts of Python programming.
- Apply the basic concepts of Python to solve simple programming problems.

Evaluation Instrument	Weightage	Learning Outcomes	Tentative dates
Assignment	25%	1, 2	In the labs
Mid term	30%	2	29-01-2024
End term	45%	1, 2	26-02-2024

Why Learn Coding?



<https://www.youtube.com/watch?v=YPE2dO5sII0>



Code stars short film

<https://www.youtube.com/watch?v=dU1xS07N-FA>

Reference materials



- Let us Python (5th Edition) by Y. Kanetkar and A. Kanetkar.
- The Complete Reference on Python by Martin C. Brown.
 - Link: <https://www.python.org/>
- https://www.tutorialspoint.com/python_data_structure/python_arrays.htm

Installation

- For windows:
 - <https://www.anaconda.com/download>
 - www.python.org/downloads (choose the appropriate one for your machine).
- For Linux:
 - `sudo apt-get install python3.8`
- To check the version:
 - `python3 --version`

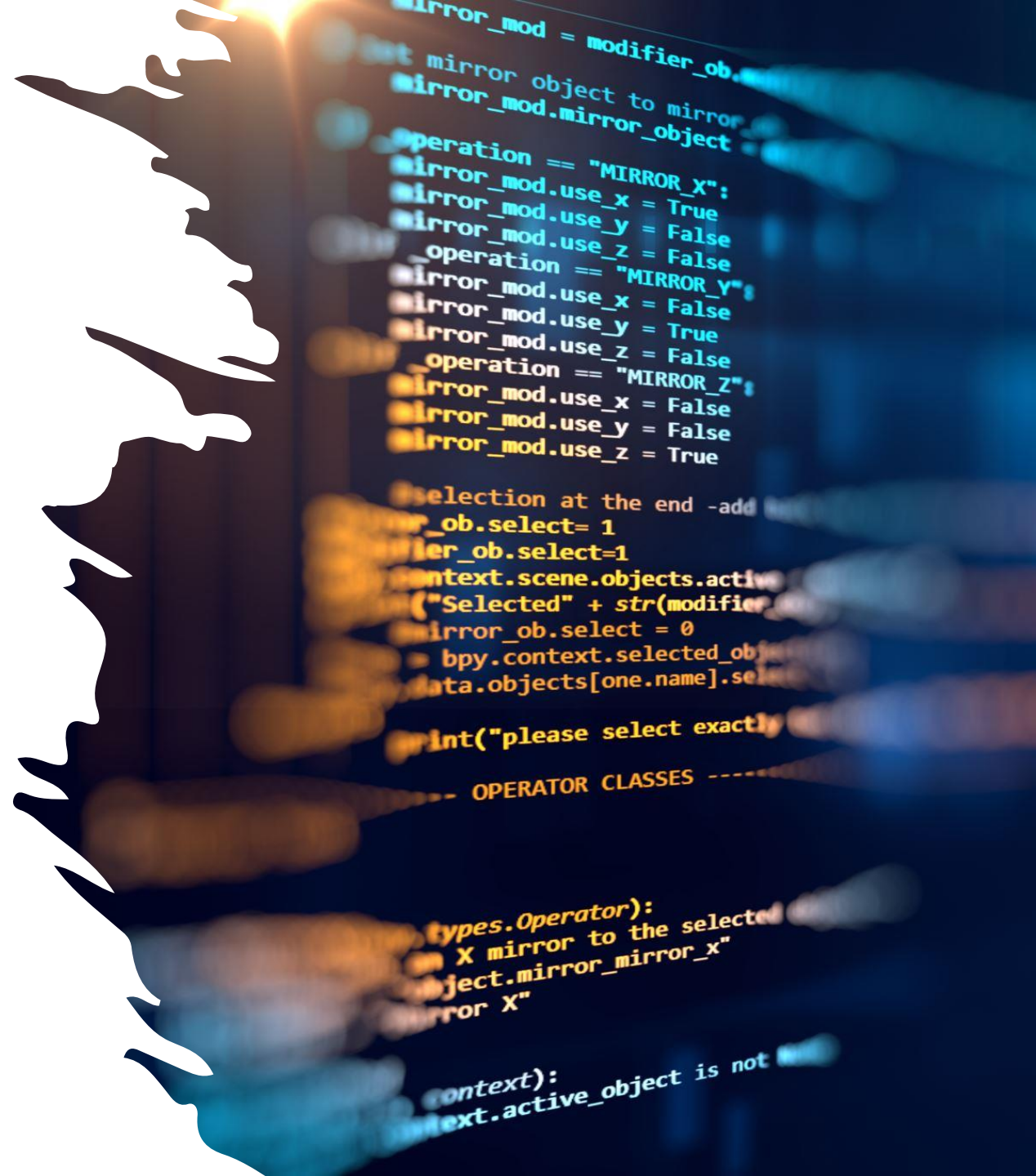


Syllabus of the course

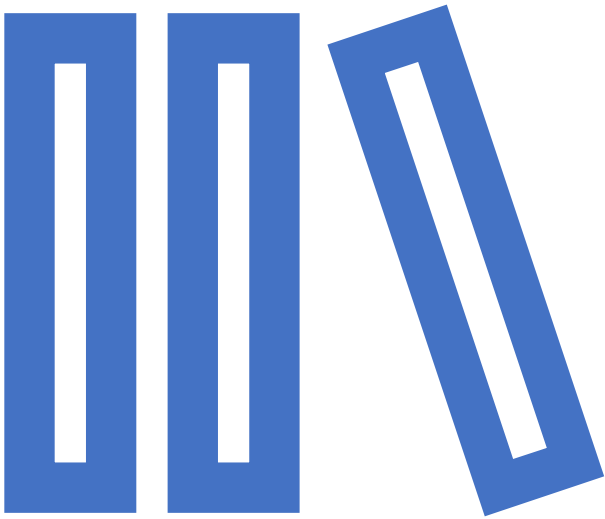
- Numbers
- String
- Dictionary
- Tuple
- List
- Modules
- Decision statements

Getting Started

- What is Python?
 - Python is a high-level programming language created by **Guido Van Rossum**.
 - Python was first released in **1991**.
 - Today Python interpreters are available for many Operating Systems including Windows and Linux.



Why Python is popular?



- Free
- Software quality: reusable and maintainable.
- Developer productivity: Less to type, debug and maintain.
- Program portability: Run unchanged on most platforms.
- Support libraries: - Strong library support for Web site construction, Numeric programming, Game development, Machine Learning etc.
- Component integration: - Can invoke C, C++ libraries and Java components
- Ease of use

Where is Python used?



System programming



Building GUI applications



Internet scripting



Component integration



Database programming



Numeric and Scientific programming



Game programming



Robotics programming

Python Marketplace



- Google - In web search system
- YouTube - Video Sharing service
- Bit-torrent - Peer to Peer file sharing system
- Intel, HP, Seagate, IBM, Qualcomm - Hardware testing
- Pixar, Industrial Light and Magic - Movie animation
- JP Morgan, Chase, UBS - Financial market forecasting
- NASA, FermiLab - Scientific programming
- iRobot - Commercial robot vacuum cleaners
- NSA - Cryptographic and Intelligence analysis
- IronPort - Email Servers

Python Specification

- Python is a specification for a language that can be implemented in different ways:
 1. Cpython: implementation in C.
 2. PyPy: Written in Rpython.
 3. Jython: Written in Java
 4. IronPython: Written in C#
- All implementations are compilers as well as interpreters.
 - The compiler converts the Python program into intermediate bytecode.
 - This bytecode is then interpreted by the interpreter.

Important packages



NumPy: Advanced mathematical operation library with support for multidimensional arrays and matrices.



SciPy: Scientific computing such as optimization, interpolation, integration etc.



Pandas: Manipulating numerical tables/time series.



Matplotlib: 2D and 3D data visualization library.



OpenCV: Open-source computer vision library.

Python virtual machine (PVM)

- PVM: A software that emulates a physical computer.
- Programs written in Python are converted into bytecode instructions by Python compiler. Bytecode instructions remain same irrespective of platform (microprocessor+ OS)
- During execution bytecode instructions are interpreted by the virtual machine (PVM) and then executed.

Getting started with Python

- Check version of Python
 - `python --version`
- First program
 - Print statement

```
print("Hello Worlds!!!!")
```

```
Hello Worlds!!!!
```

Taking Input from User

```
a=input()
```

```
b=input()
```

```
print(a,b)
```

```
Hello
```

```
World !!!
```

```
Hello World !!!
```

Errors and their types

Error is the flaw available in your code which may stop the compilation (it will not be able to run) or results in some wrong input. It is categorized in 3 parts:

- a. Compile time Error
- b. Run time Error
- c. Logical Error

Compile time error

- At the time of compilation, the system checks the source code.
- For any fault or violation of programming convention or rules, compilation stops, and system throws compilation error.
- Types:
 - Syntax error
 - Semantic error



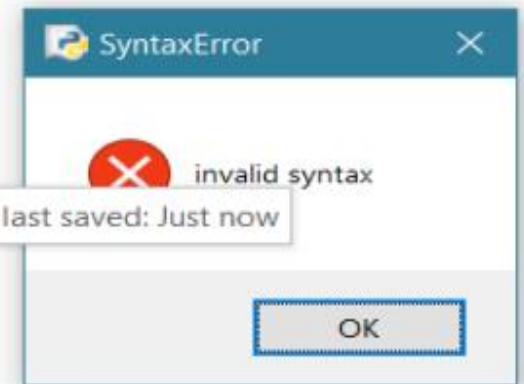
Compilation Error: Types

- **Syntax Error:**

- Syntax refers to the formal rules those are used for writing valid statements in a programming language, if compiler finds any difference between the formal rule and the code written by coder, then it produces error.

File Edit Format Run Options Window Help

```
a=input() :  
b=input()  
print(a,b)
```

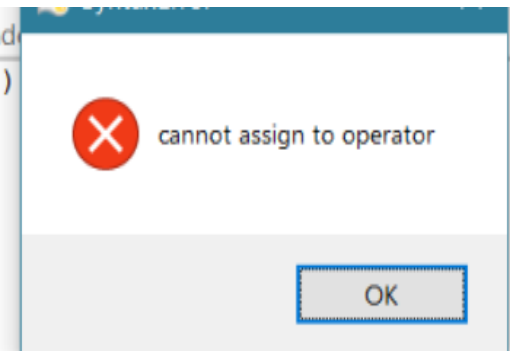


- **Semantics Error:**

- Semantic refers to the meaning of a statement. So, when a statement doesn't make any sense and is not meaningful then we can say it is a Semantic error.
- Also known as "Bugs" and are found during the process of debugging.

File Edit Format Run Options Wind

```
a=int(input("Enter Number"))  
sum=0  
sum+a=sum
```



Runtime Error

- This type of errors occurs after the compilation of the code; in-between when execution is going on. Due to when program is “crashed” or “abnormally abrupted”.

Examples:

- Infinite loop
- Wrong value as Input
- Invalid function call
- Divide by Zero

Logical error

- Sometimes programmer's analysis of the problem is wrong, in those cases the errors are logical errors.
- For example,
 - Incorrect implementation of an algorithm
 - Unmarked end of loop
 - Wrong parameters passed

Program:

```
x = float(input('Enter a number: '))  
y = float(input('Enter a number: '))  
z = x+y/2  
print ('The result is:', z)
```


Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Python uses indentation to indicate a block of code.

```
if 14 > 9:  
    print("Fourteen is greater than nine!")
```

Fourteen is greater than nine!

```
if 14 > 9:  
    print("Fourteen is greater than nine!")
```

Fourteen is greater than nine!

```
if 14 > 9:  
print("Fourteen is greater than nine!")
```

Cell In[7], line 2

```
    print("Fourteen is greater than nine!")  
    ^
```

IndentationError: expected an indented block after 'if' statement on line 1

Python Comments

- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.

```
#Single line Comment  
#This is a comment  
print("Hello, World!")
```

```
#Multi line comments  
"""  
This is a comment  
written in  
more than just one line  
"""  
print("Hello, World!")
```