



**SNS COLLEGE OF ENGINEERING**  
Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



**DEPARTMENT OF CSE (IoT & CYBER SECURITY INCLUDING BLOCKCHAIN TECHNOLOGY)**



# **19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING**

❖ A readable, dynamic, pleasant, flexible, fast and powerful language

# UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS

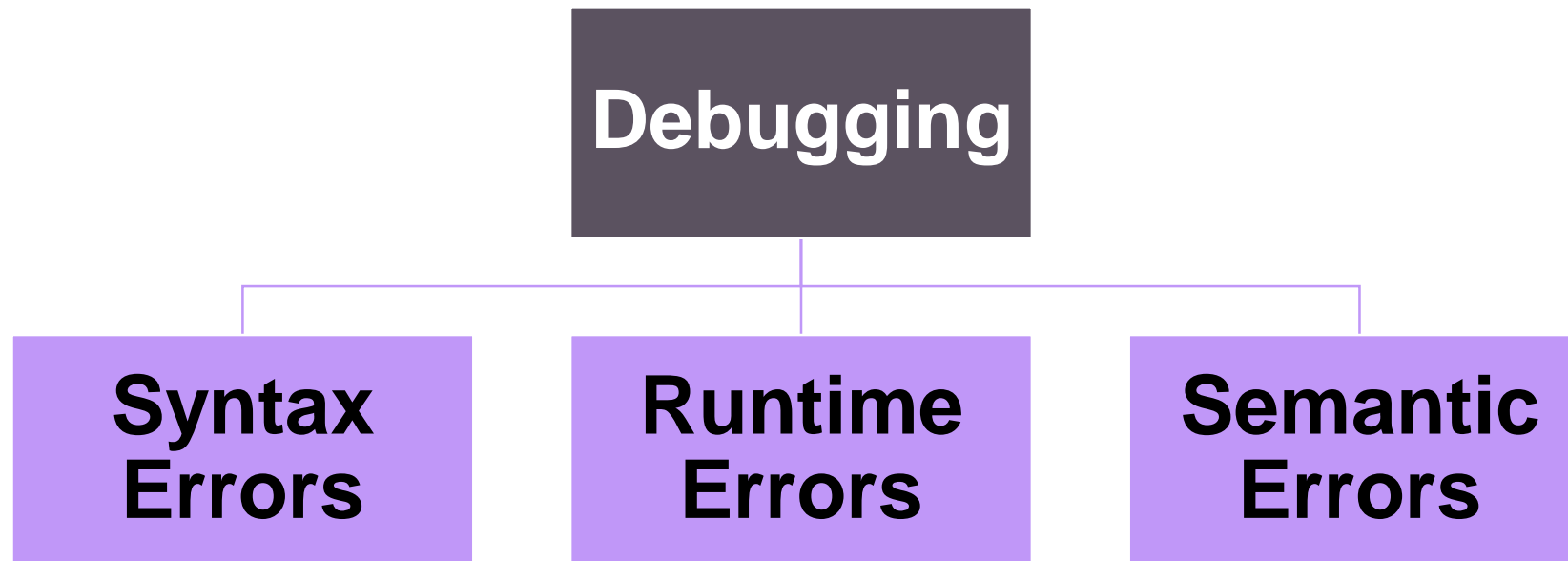
- Python interpreter and interactive mode, **debugging**; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

# Recap

- **Compiler:** A compiler is a program that translates source code into object code to be understood by a specific central processing unit (CPU).
- **Interpreter:** An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.
- Features of Python
- Python Interpreter
  - Interactive Mode
  - Script Mode

# Debugging

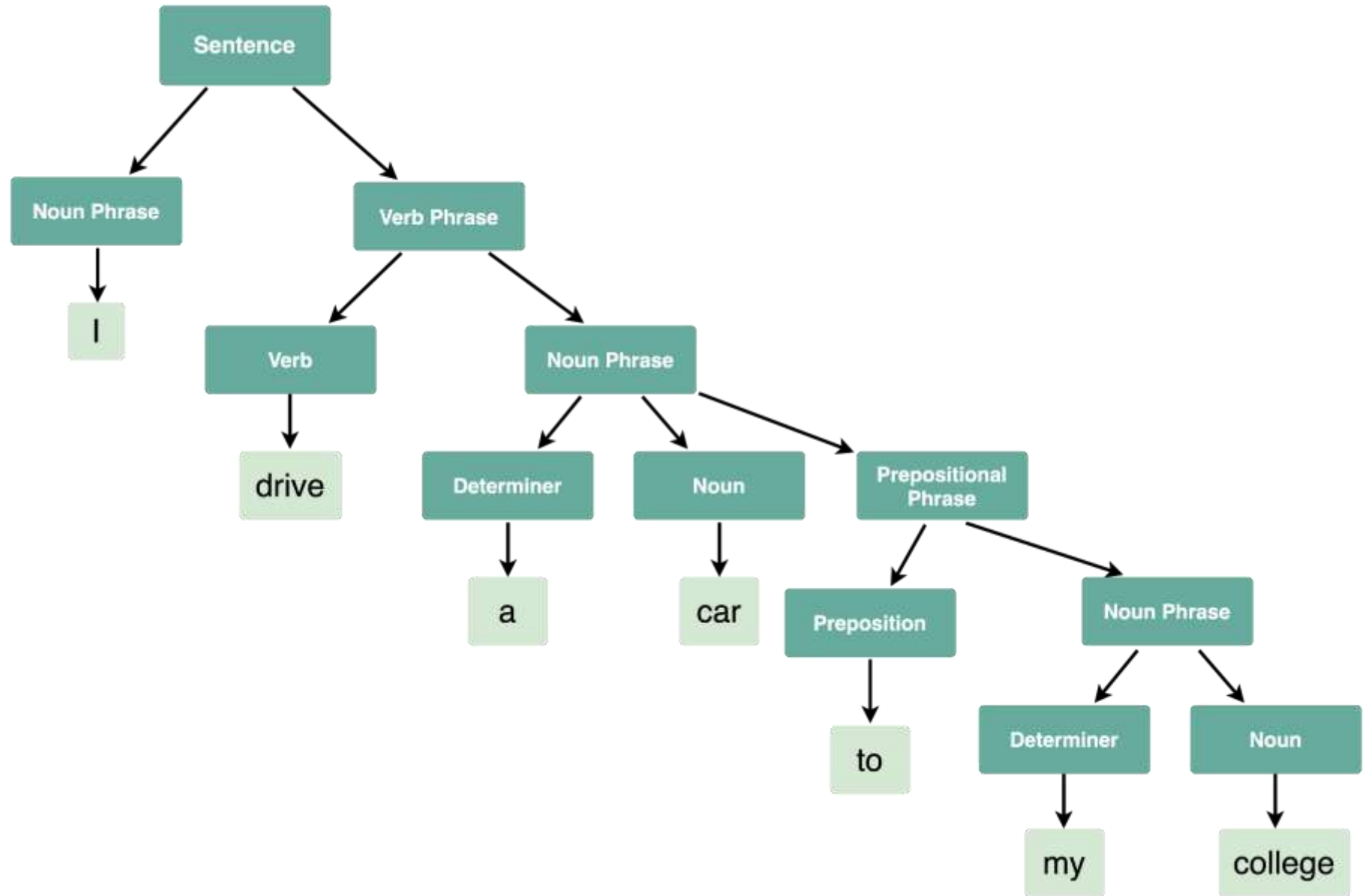
- Programming is error-prone.
- Programming errors are called bugs and the process of tracking them down is called debugging.



# Debugging



# Syntax



# Syntax

- Syntax refers to the rules that define the structure of a language.
- Syntax in computer programming means the rules that control the structure of the symbols, punctuation, and words of a programming language.
- If the syntax of a language is not followed, the code will not be understood by a compiler or interpreter.

# Syntax errors

- Python can only execute a program if the syntax is correct; otherwise, the interpreter displays an error message.
- Every language has its own set of rules that make up its basic syntax.
- For example, parentheses have to come in matching pairs, so  $(1 + 2)$  is legal, but  $8)$  is a syntax error.



# Syntax errors

```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> (1+2)
3
>>> 8)
SyntaxError: unmatched ')'
```

```
>>> print(hello)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print(hello)
NameError: name 'hello' is not defined
>>>
```

# Runtime errors

- This error does not appear until after the program has started running.
- These errors are also called exceptions because they usually indicate that something exceptional (and bad) has happened.
- Here are some examples of common runtime errors you are sure to encounter:
  - Misspelled or incorrectly capitalized variable and function names
  - Attempts to perform operations (such as math operations) on data of the wrong type (ex. attempting to subtract two variables that hold string values)
  - Dividing by zero
  - Attempts to use a type conversion function such as **int** on a value that can't be converted to an **int**.

# Runtime errors

tax.py - C:/Users/ambika sathish/AppData/Local/Programs/Py

File Edit Format Run Options Window Help

```
subtotal = input("Enter total before tax:")
tax = .08 * subTotal
print("tax on", subtotal, "is:", tax)
```

Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win3

2

Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: C:/Users/ambika sathish/AppData/Local/Programs/Python/Python38/tax.py

Enter total before tax: 5000

Traceback (most recent call last):

File "C:/Users/ambika sathish/AppData/Local/Programs/Python/Python38/tax.py", line 2, in <module>

tax = .08 \* subTotal

NameError: name 'subTotal' is not defined

>>> |

# Semantic errors

- The third type of error is the semantic error.
- If there is a semantic error in your program, it will run successfully in the sense that the computer will not generate any error messages, but it will not do the right thing.
- The problem is that the program you wrote is not the program you wanted to write.
- Identifying semantic errors can be tricky because it requires you to work backward by looking at the output of the program and trying to figure out what it is doing.

# Semantic errors

sum.py - C:/Users/ambika sathish/AppData/Local/Programs/Python/Pyt

File Edit Format Run Options Window Help

```
num1 = input('Enter a number:')  
num2 = input('Enter another number:')  
sum = num1 + num2  
  
print('The sum of', num1, 'and', num2, 'is', sum)
```

Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win3

2

Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: C:/Users/ambika sathish/AppData/Local/Programs/Python/Python38/sum.py

Enter a number:10

Enter another number:20

The sum of 10 and 20 is 1020

>>> |