**Python Introduction - (2h)**

1. Who made python and Who maintains python and Where is the source?
2. The Python Community, PyCon, PyPI & PEP.
3. Where is Python being used ?
4. Advantages & Drawbacks of Python.
5. Different Runtimes of Python - Jython, Cython, Brython.
6. Different Flavors of Python - Spyder, Anaconda, IPython, PyPy.

**Python Installation - (1-2h)**

1. On Windows
2. On Linux.
3. On Mac.

**Python Fundamentals - (30h)**

1. **How to run python (2h)**
    a. Philosophy of Python.
    b. Python Interpreter.
    c. Running Python Scripts.
    d. Compiling Python.
2. **Python Essentials (1h)**
    a. Source Code Encoding.
    b. What is Class and Object, What are methods and data members of a class?
    c. Everything is an Object in Python.
    d. *dir* and *help* function.
    e. The *id* function
3. **Programming related fundamentals (2-3h)**
    a. Comments
    b. Keywords (*mention keyword module*)
    c. Identifiers (*mention special identifiers e.g. _(underscore)*)
    d. Literals (Bytes & ByteArray, String, Numbers, Sets, Lists, Tuples & Dictionary)
    e. Operators (*Arithmetic, Relational, Logical, Bitwise, Assignment*)
    f. Character Encoding and Decoding. (*utf-8, ascii, unicode, chr, ord*)
    g. Standard Input Output (*stdin, stdout, stderr*)

4. **Flow control in Python (2-3h)**
    a. Conditionals-
        i. if , if-else, if-elif
        ii. Nesting of Conditionals
    b. Looping and Iteration-
        i. for - in
        ii. Enumeration
        iii. While
        iv. Loop - else

v. Break, continue, pass
vi. Nesting of loops.

5. **Functions & modules in Python (2-3h)**
    a. Inbuilt python functions.
    b. Creating a user defined function.
    c. __name__
    d. Function docstring and __doc__
    e. Invoking a function.
    f. Function Arguments.
    g. Lambda Functions.
    h. Function call in python - By reference or By value?
    i. Creating Python modules
    j. Importing modules
    k. Installing external modules using pip, easy_install
    l. Scope of variables (locals & globals)

6. **Data Structures using Python (13h)**
    a. Immutable and Mutable Data structures. (1/2h)
    b. Iterators & Generators
    c. Numbers - (1h)
        i. Int
        ii. float & NaN
    d. Strings (1.5h)
        i. Creation
        ii. ord & chr
        iii. String Operations
        iv. String Functions & Methods
        v. String Formatting
    e. Lists (3h)
        i. Creation
        ii. Adding an item - Insert, Append, Extend
        iii. Deleting an item - Remove, Pop, Clear, del
        iv. Indexing an item - index, count, *list slicing*
        v. Sorting a List - sort, reverse
        vi. Copying a list
        vii. List Comprehension.
    f. Tuples (2h)
        i. Creation
        ii. Indexing an item - index, count, *tuple slicing*
        iii. Copying a tuple
        iv. Concatenating two Tuples
        v. Comparing Tuples
        vi. Tuple Comprehension
    g. Sets (2h)

i. Creation
ii. Set Operations - Union, Intersection, Difference
iii. Updating a set
iv. Set Comprehension

h. Dictionaries (3h)
i. Creation
ii. Keys & Values operations
iii. Iterating over a dictionary values
iv. Dictionary Comprehension

**7. File Handling (2h)**

a. Read a file.
b. Create/ Write a file.
c. Append a file.
d. Seek and Tell in a file

**8. Errors & Exceptions (2h)**

a. SyntaxError
b. Some Common Semantic Errors -
i. ZeroDivisionError
ii. NameError
iii. TypeError
iv. ValueError
c. try, except, finally.
d. Raising Custom Exceptions

**9. Object Oriented Programming with Python (5h)**

a. The distinction between Algorithmic and Object Oriented Approach ()
b. Theoretical basics of Object Oriented Programming-
i. Data Abstraction
ii. Inheritance & Multiple Inheritance
iii. Polymorphism
iv. Data Hiding (double underscored properties)
v. Data Encapsulation
c. A Python Class
d. Methods & Data Members in Classes
e. @staticmethod & @classmethod
f. Special Methods of Classes
i. __init__
ii. __str__
iii. __repr__
iv. __del__
g. isinstance method

**Total hours - 34**

**Pre-requisites -**

C & a bit of Java or any other OOP (*optional*).