PIUPY

WHERE YOU WRITE YOUR SUCCESS !!!

Give us a chance because YOU DESERVE THE BEST !!!

11th Class Syllabus (Computer Science using Python)

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>   All Modules are mandatory   <<<<<<<<<<<<<<<<<<<<<<<<<<<<<

Module 1 : (Basics of Python)

- Features and uses of Python
- Datatypes (int, str, float, bool, complex)
- Keywords and use of ‘None’ keyword
- Identifiers and their rules
- Comments (Single Line (#) and Multi Line (using Function documentation format ‘’’   ‘’’))
- Multi-Line Strings (using \ and ‘’’ ‘’’), escape sequences, Raw strings (r’ ‘ or R’ ‘)

Module 2 : (Getting Started with Coding in Python)

- Input and output (input( ) and print( ))
- Ways to print (f-string, c-style (tuple format), format( ), normal style)
- Operators
  - Logical (and, or, not)
  - Arithmetic (+, -, *, /, //, %, **)
  - Relational (==, !=, <, <=, >, >=)
  - Assignment (=) and Shorthand (+=, -=, *=, /=, //=, %=, **=)
  - Membership (in, not in)
  - Identity (is, is not)
  - Shift (<<, >>)
  - Bitwise (&, |, ^)

- Dynamic Typing in Python
- Printing in a single line (end=’’)

- Difference between end=’’ and sep=’’ in print( )
- Mutable types (list, dict, set) and Immutable Types (int, str, float, bool, complex, tuple)
- Use of id( ), bin( ), oct( ), hex( ), chr( ), ord( )
- Random numbers
    - random.random( )   (0 <= number < 1)
    - random.randint(10,20)   (10 <= number <= 20)
    - random.randrange(20)   (0 <= number < 20)
    - random.randrange(10,20)   (10 <= number < 20)
    - random.randrange(10,31,5)   (10 <= number < 31 i.e. [10, 15, 20, 25, 30])

<u>Module 3 : (Control Statements and Iteration)</u>

- Conditional Statements
    - If block (if( ):)
    - If-else block (if( ):   else:)
    - If-else ladder (if( ):   elif( ): …..   else:)
    - Nested if-else statements
    - Shortcut of using if-else (result1 if (condition_found_true) else result2)
- Iterative statements
    - Use of range(start, end+1, step)
    - for Loop
    - while Loop
    - Implementing do-while Loop using while Loop
    - ‘break’ statement
    - ‘continue’ statement
    - Nested Loops
    - Use of ‘else:’ in Loops
    - Pattern Programming (using special characters, alphabets and numbers)
    - Program to check for Leap year
    - Program to check for Prime Number
    - Program to find the sum of the digits of a given number
    - Program to check for Armstrong Number
    - Program to check for Perfect Number
    - Program to check for Palindrome Numbers as well as Palindrome Strings
    - Program to print Fibonacii Series

<u>Module 4 : (Collections in Python)</u>

- **String**
    - Operators used on Strings
    - String Replication
    - String Slicing [: :]
    - Methods used on Strings like capitalize( ), find( ), replace( ), isalnum( ), isalpha( ), isdigit( ), isspace( ), isupper( ), islower( ), upper( ), lower( ), strip( ), lstrip( ), rstrip( )
    - Use of enumerate( ) [index provider] in Strings

- **List**
    - Creating an empty list (using list( ) or [ ])
    - Creating a list having elements (using [ ])
    - Creating nested lists ([, , , [ ] ,)
    - Reading a list (using eval(input( )))
    - Accessing a list (using indexing)
    - Use of enumerate( ) [index provider] in Lists
    - List slicing, traversing, replication, joining
    - List functions like index( ), append( ), extend( ), insert( ), pop( ), remove( ), clear( ), count( ), reverse( ), sort( ), sort(reverse = True)
    - Shallow copy and deep copy of list
    - Global functions like del, max( ), min( ), sum( ), len( )
    - Typecasting from and into a List (using list( ))
    - Deleting a list

- **Tuple**
    - Creating an empty tuple (using tuple( ) or ( ))
    - Creating a tuple with single element (using (value,))
    - Creating a tuple having elements (using ( ))
    - Creating nested tuples ((, , , ( ) ,))
    - Reading a tuple (using eval(input( )))
    - Accessing a tuple (using indexing)
    - Use of enumerate( ) [index provider] in Tuples
    - Tuple slicing, traversing, replication, joining
    - Tuple functions like index( ), count( )
    - Shallow copy and deep copy of tuple
    - Global functions like del, max( ), min( ), sum( ), len( )
    - Unpacking of tuples
    - Typecasting from and into a Tuple (using tuple( ))
    - Modifying a tuple (directly and indirectly)
    - Deleting a tuple

- **Dictionary**
    - Creating an empty dictionary (using dict( ) or { })
    - Creating a dictionary having elements (using { : , : })
    - Creating nested dictionaries ({ : , : { } })
    - Reading a dictionary (using eval(input( )))
    - Accessing a dictionary (using keys)
    - Use of enumerate( ) [index provider] in Dictionaries
    - Use of key( ), values( ) and items( )
    - Dictionary traversing
    - Use of zip( )
    - Adding, updating and deleting (using del or pop( )) elements
    - Checking for existence of a key or a value (using ‘in’ operator)
    - Printing a dictionary in json format (using json.dumps(dct, indent=2))
    - Dictionary functions like clear( ), get( ), key( ), values( ) and items( ), update( )
    - Typecasting from and into a Tuple (using dict( ))
    - Shallow copy and deep copy of dictionary

    - Global functions like del, max( ), min( ), sum( ), len( )
    - Deleting a dictionary

- **Set**
    - Creating an empty set (using set( ))
    - Creating a set having elements (using { , , })
    - Creating nested sets ({ , , { }, })
    - Reading a set (using eval(input( )))
    - Use of enumerate( ) [index provider] in Sets
    - Set traversing
    - Set functions like add( ), pop( ), remove( ), discard( ), clear( ), union( ) or '|', intersection( ) or '&', difference( ) or '-', symmetric_difference( ) or '^' (which is 'union – intersection'), update( )
- Use of split( ) and join( ) in collections (lst.split(',') or ','.join(lst) where 'lst' is any collection)

<u>Module 5 : (Exception Handling and Sorting Algorithms)</u>

Exception Handling

- Types of Errors (Syntax, Semantic, Logical, Runtime)
- Exception handling (using try, except and finally)
- Printing our custom error message as well as system generated error message
- Raising our own exception (using raise Exception('Error Message'))
- Debugging a Program

Sorting Algorithms

- Bubble Sort
- Insertion Sort

<u>Module 6 : (Introduction to SQL)</u>

- Features and Uses of SQL
- Degree and Cardinality
- Datatypes (int, float, double, char, varchar, date)
- Operators
    - Logical (and, or, not)
    - Arithmetic (+, -, *, /, % or mod(x,y))
    - Relational (==, !=, < >, <, <=, >, >=)
- Control statements (using 'case' both for implementing 'if-else' (after) as well as 'switch' (before))
- Constraints
    - Primary key
    - Foreign key
    - Unique
    - Check
    - Not null
    - Default

Module 7 : (Getting Started with SQL commands)

- DDL commands
  - Create
  - Alter (adding, modifying, renaming and dropping a column)
  - Drop
  - Rename
  - Truncate
- DRL / DQL commands
  - Select
- DML commands
  - Insert
  - Update
  - Delete
- DCL commands
  - Grant
  - Revoke
- TCL commands
  - Commit
  - Rollback
  - Savepoint

Module 8 : (Understanding SQL clauses)

- Aggregate functions
- 'distinct' clause
- 'order by' clause
- 'in' operator
- 'like' operator
- ifnull( )
- Subqueries
- Set operators
  - union
  - union all
  - intersection (using 'in')
  - minus (using 'not in')
- 'group by' clause
- 'having' clause
- Difference between 'where' and 'having' clauses

Module 9 : (Learning SQL Joins)

- Joins
  - Inner Join
    - Equi Join
    - Non-Equi Join
  - Outer Join
    - Left Outer Join

        - Right Outer Join
        - Full Outer Join
    - Cross Join
    - Natural Join
    - Self Join

Module 10 : (Using SQL built-in functions and implementing them in SQL queries)

- Built-in functions
    - length( )
    - lower( ) or lcase( )
    - upper( ) or ucase( )
    - left( )
    - right( )
    - mid( ) or substr( ) or substring( )
    - ltrim( )
    - rtrim( )
    - trim( )
    - power( ) or pow( )
    - concat( )
    - current_date( ) or curdate( )
    - current_time( ) or curtime( )
    - now( )
    - sysdate( )
    - sleep( )
    - day( )
    - month( )
    - year( )
    - dayname( )
    - monthname( )
    - round( )
    - truncate( )

Module 11 : (Introduction to NOSQL using MongoDb and Basics of Cyber Security)

- CRUD operations in MongoDB
    - Creating Collections
    - Inserting documents
    - Updating documents
    - Deleting documents
- Cyber Security
    - Awareness about Cyber security threats and attacks
    - Staying protected in Cyber world
    - Basics of Ethical Hacking

>>>>> **END of Syllabus** <<<<<