UNIT I - COMPUTATIONAL THINKING AND ALGORITHMIC PROBLEM SOLVING

PART- A (2 Marks)

1. **What is an algorithm?(Jan-2018)**
   Algorithm is an ordered sequence of finite, well defined, unambiguous instructions for completing a task. It is an English-like representation of the logic which is used to solve the problem. It is a step-by-step procedure for solving a task or a problem. The steps must be ordered, unambiguous and finite in number.
2. **Write an algorithm to find minimum of three numbers.**
   ALGORITHM : Find Minimum of three numbers
   Step 1: Start
   Step 2: Read the three numbers A, B, C
   Step 3: Compare A,B and A,C. If A is minimum, perform step 4 else perform step 5.
   Step 4: Compare B and C. If B is minimum, output "B is minimum" else output "C is minimum".
   Step 5: Stop
3. **List the building blocks of algorithm.**
   The building blocks of an algorithm are
   - Statements
   - Sequence
   - Selection or Conditional
   - Repetition or Control flow
   - Functions

   An action is one or more instructions that the computer performs in sequential order (from first to last). A decision is making a choice among several actions. A loop is one or more instructions that the computer performs repeatedly.
4. **Define statement. List its types.**
   The instructions in Python, or indeed in any high-level language, are designed as components for algorithmic problem solving, rather than as one-to-one translations of the underlying machine language instruction set of the computer. There are three types of high-level programming language statements. Input/output statements make up one type of statement. An input statement collects a specific value from the user for a variable within the program. An output statement writes a message or the value of a program variable to the user's screen.
5. **Write the pseudocode to calculate the sum and product of two numbers and display it.**
   INITIALIZE variables sum, product, number1, number2 of type real
   PRINT "Input two numbers"
   READ number1, number2
   COMPUTE sum = number1 + number2
   PRINT "The sum is", sum
   COMPUTE product = number1 * number2
   PRINT "The Product is", product
   END program
6. **How does flow of control work?**
   Control flow (or flow of control) is the order in which individual statements, instructions or function calls of an imperative program are executed or evaluated. A control flow statement is a statement in which execution results in a choice being made as to which of two or more paths to follow.
7. **Write the algorithm to calculate the average of three numbers and display it.**
   Step 1: Start
   Step 2: Read values of X,Y,Z
   Step 3: S = X+Y+Z
   Step 4: A = S/3
   Step 5: Write value of A
   Step 6: Stop
8. **Give the rules for writing Pseudocode.**
   - Write one statement per line.
   - Capitalize initial keywords.
   - Indent to show hierarchy.
   - End multiline structure.
   - Keep statements language independent.

9. What is a function?

Functions are named sequence of statements that accomplish a specific task. Functions usually "take in" data, process it, and "return" a result. Once a function is written, it can be used over and over and over again. Functions can be "called" from the inside of other functions.

10. Give the difference between flowchart and pseudocode.

Flowchart	Pseudocode
• A flowchart is a diagram showing an overview of the problem. • It is a pictorial representation of how the program will work, and it follows a standard format. • It uses different kinds of shapes to signify different processes involved in the problem	• Pseudocode is a means of expressing the stepwise instructions for solving a problem without worrying about the syntax of a particular programming language. • Unlike a flowchart, it uses a written format which requires no absolute rules for writing. • It can be written in ordinary English, and we can use some keywords in it too.

11. Define a flowchart.

- A flowchart is a diagrammatic representation of the logic for solving a task.
- A flowchart isdrawn using boxes of different shapes with lines connecting them to show the flow of control.
- The purpose of drawing a flowchart is to make the logic of the program clearer in a visual form.

12. Give an example of iteration.

```
a = 0
for i in range(4):          # i takes the value 0,1,2,3
        a = a + i
print(a)                    #number 6 is printed (0+0;0+1;1+2;3+3)
```

13. Write down the rules for preparing a flowchart.

While drawing a flowchart, some rules need to be followed:

- A flowchart should have a start and end
- The direction of flow in a flowchart must be from top to bottom and left to right
- The relevant symbols must be used while drawing a flowchart.

14. List the categories of Programming languages.

Programming Languages are divided into the following categories:

- Interpreted
- Functional
- Compiled
- Procedural
- Scripting
- Markup
- Logic-Based
- Concurrent
- Object-Oriented Programming Languages.

15. Mention the characteristics of algorithm.

- Algorithm should be precise and unambiguous.
- Instruction in an algorithm should not be repeated infinitely.
- Ensure that the algorithm will ultimately terminate.
- Algorithm should be written in sequence.
- Algorithm should be written in normal English.
- Desired result should be obtained only after the algorithm terminates

16. List out the simple strategies to develop an algorithm.

Algorithm development process consists of five major steps.

Step 1: Obtain a description of the problem.

Step 2: Analyze the problem.

Step 3: Develop a high-level algorithm.

Step 4: Refine the algorithm by adding more detail.

Step 5: Review the algorithm.

17. Compare machine language, assembly language and high-level language.

Machine Language	Assembly Language	High-Level Language
• The language of 0s and 1s is called as machine language. • The machine language is system independent because there are different set of binary instruction for different types of computer systems	• It is low level programming language in which the sequence of 0s and 1s are replaced by mnemonic (ni-monic) codes. • Typical instruction for addition and subtraction	• High level languages are English like statements and programs. • Written in these languages are needed to be translated into machine language before to their execution using a system software compiler.

18. Describe algorithmic problem solving.

Algorithmic Problem Solving deals with the implementation and application of algorithms to a variety of problems. When solving a problem, choosing the right approach is often the key to arriving at the best solution.

19. Give the difference between recursion and iteration.

Recursion	Iteration
Function calls itself until the base condition is reached.	Repetition of process until the condition fails.
In recursive function, base case (terminate condition) and recursive case are specified.	Iterative approach involves four steps: initialization, condition, execution and updation.
Recursion is slower than iteration due to overhead of maintaining stack.	Iteration is faster.
Recursion takes more memory than iteration due to overhead of maintaining stack.	Iteration takes less memory.

20. What are advantages and disadvantages of recursion?

Advantages	Disadvantages
Recursive functions make the code look clean and elegant.	Sometimes the logic behind recursion is hard to follow through.
A complex task can be broken down into simpler sub-problems using recursion.	Recursive calls are expensive (inefficient) as they take up a lot of memory and time.
Sequence generation is easier with recursion than using some nested iteration.	Recursive functions are hard to debug.

21. Write an algorithm to accept two numbers, compute the sum and print the result.(Jan-2018)

Step1: Read the two numbers a and b.
Step 2: Calculate sum = a+b
Step 3: Display the sum

22. Write an algorithm to find the sum of digits of a number and display it.

Step 1: Start
Step 2: Read value of N
Step 3: Sum = 0
Step 4: While (N != 0)

```
Rem = N % 10
Sum = Sum + Rem
N = N / 10
```

Step 5: Print Sum
Step 6: Stop

23. Write an algorithm to find the square and cube and display it.

Step 1: Start
Step 2: Read value of N
Step 3: S =N*N
Step 4: C =S*N
Step 5: Write values of S,C
Step 6: Stop

24. Explain Tower of Hanoi.

The Tower of Hanoi is a mathematical game. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
- No disk may be placed on top of a smaller disk.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks.

25. What is recursion?

Recursion is a method of solving problems that involves breaking a problem down into smaller and smaller subproblems until you get to a small enough problem that it can be solved trivially. Usually recursion involves a function calling itself. While it may not seem like much on the surface, recursion allows us to write elegant solutions to problems that may otherwise be very difficult to program.

Example:

```
defcalc_factorial(x):
        if x == 1:
                return 1
        else:
                return (x * calc_factorial(x-1))
num = 4
print("The factorial of", num, "is", calc_factorial(num))
```

26. Write an algorithm to find minimum in a list. (Jan-2019)

ALGORITHM : To find minimum in a list

Step 1: Start
Step 2: Read the list

Step 3: Assume the first element as minimum
Step 4: Compare every element with minimum. If the value is less than minimum, reassign that value as minimum.
Step 5: Print the value of minimum.
Step 6: Stop

27. Distinguish between algorithm and program. (Jan-2019)

Algorithm	Program
Algorithm is the approach / idea to solve some problem.	A program is a set of instructions for the computer to follow.
It does not have a specific syntax like any of the programming languages	It is exact code written for problem following all the rules (syntax) of the programming language.
It cannot be executed on a computer.	It can be executed on a computer

28. List the Symbols used in drawing the flowcart. (May 2019)

Flowcharts are usually drawn using some standard symbols

- Terminator
- Process
- Decision
- Connector
- Data
- Delay
- Arrow

29. Give the python code to find the minimum among the list of 10 numbers. (May 2019)

```
numList = []
n=int(raw_input('Enter The Number of Elements in List :'))
for i in range(0, n):
        x = raw_input('Enter the Element %d :' %(i+1))
        numList.append(x)
maxNum = numList[0]
for i in numList:
      if i > maxNum:
            maxNum = i
print('Maximum Element of the Given List is %d' %(int(maxNum)))
```

PART B & C (16 MARKS)

1. What are the building blocks of an algorithm? Explain in detail.
2. Briefly describe iteration and recursion. Explain with algorithm.
3. Explain Algorithmic problem solving.
4. Write an algorithm and draw a flowchart to calculate 2^4.
5. Illustrate Algorithm, Psuedocode and Flowchart with an example.
6. a) Describe Psuedocode with its guidelines.
   b) Give an example for psuedocode.
   c) Write the pseudocode for Towers of Hanoi.
7. a) What is flowchart?
   b) List down symbols and rules for writing flowchart.
   c) Draw a flowchart to count and print from1 to 10.
8. a) Write a program to find the net salary of an employee.
   b) Write a program to guess an integer number in a range.
9. a) Write a program to insert a card in a list of sorted cards. **(Jan-2019)**
   b) Write a program to find the minimum number in a list.
10. a) Draw a flow chart to accept three distinct numbers, find the greatest and print the result. **(8) (Jan-2018)**
    b) Draw a flow chart to find the sum of the series 1+2+3+……+100. **(8)(Jan-2018)**
11. Outline the Towers of Hanoi problem. Suggest a solution to the Towers of Hanoi problem with relevant diagrams. **(16) (Jan-2018)**
12. Identify simple strategies for developing an algorithm. **(Jan-2019)**
13. Mention the different types of iterative structures allowed in python. Explain the use of continue and break statements with an example. **(May 2019)**
14. (a) What is an algorithm? Summarise the characteristics of a good algorithm. **(8) (May 2019)**
    (b) Outline the algorithm for displaying the first n odd numbers. **(May 2019)**

UNIT II – DATA TYPES, EXPRESSIONS, STATEMENTS

PART- A (2 Marks)

1. **What is meant by interpreter?**
   An interpreter is a computer program that executes instructions written in a programming language. It can eitherexecute the source code directly or translate the source code in a first step into a more efficient representation and executes this code.
2. **How will you invoke the python interactive interpreter?**
   The Python interpreter can be invoked by typing the command "python" without any parameter followed by the "return" key at the shell prompt.
3. **What are the commands that are used to exit from the python interpreter in UNIX and windows?**
   CTRL+D is used to exit from the python interpreter in UNIX and CTRL+Z is used to exit from the python interpreter in windows.
4. **Define a variable and write down the rules for naming a variable.**
   A name that refers to a value is a variable .Variable names can be arbitrarily long. They can contain both letters and numbers, but they have to begin with a letter. It is legal to use uppercase letters, but it is good to begin variable names with a lowercase letter.
5. **Write a snippet to display "Hello World" in python interpreter.**
   In script mode:
   >>>print("Hello World")
   Hello World
   In Interactive Mode:
   >>> "Hello World"
   'Hello World'
6. **List down the basic data types in Python.**
   - Numbers
   - String
   - List
   - Tuple
   - Dictionary

7. **Define keyword and enumerate some of the keywords in Python. (Jan-2019)**
   A keyword is a reserved word that is used by the compiler to parse a program. Keywords cannot be used as variable names. Some of the keywords used in python are:
   - and
   - del
   - from
   - not
   - while
   - is
   - continue
8. **What do you mean by an operand and an operator? Illustrate your answer with relevant example.**
   An operator is a symbol that specifies an operation to be performed on the operands. The data items that an operator acts upon are called operands. The operators +, -, *, / and ** perform addition, subtraction, multiplication, division and exponentiation.
   Example: 20+32.In this example, 20 and 32 are operands and + is an operator.
9. **Explain the concept of floor division.**
   The operation that divides two numbers and chops off the fraction part is known as floor division.
   Example:>>> 5//2= 2
10. **Define an expression with example.**
    An expression is a combination of values, variables, and operators. An expression is evaluated using assignment operator.**Example:**Y= X + 17
11. **Define statement and mention the difference between statement and an expression.**
    A statement is a unit of code that the Python interpreter can execute. The important difference is that an expression has a value but a statement does not have a value.
12. **What is meant by rule of precedence? Give the order of precedence.**

The set of rules that govern the order in which expressions involving multiple operators and operands are evaluated is known as rule of precedence. Parentheses have the highest precedence followed by exponentiation. Multiplication and division have the next highest precedence followed by addition and subtraction.

13. Illustrate the use of * and + operators in string with example.

The * operator performs repetition on strings and the + operator performs concatenation on strings.

Example:>>> 'Hello*3'

Output: HelloHelloHello

>>>'Hello+World'

Output: HelloWorld

14. What is function call?

A function is a named sequence of statements that performs a computation. When you define a function, you specify the name and the sequence of statements. Later, you can "call" the function by name is called function call.

Example:

```
sum()       //sum is the function name
```

15. What is a local variable?

A variable defined inside a function. A local variable can only be used inside its function.

Example:

```
deff():
        s = "Me too."    // local variable
        print(s)
a = "I hate spam."
f()
print (a)
```

16. Define arguments and parameter.

A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function. Inside the function, the arguments are assigned to variables called parameters.

17. What do you mean by flow of execution?

- In order to ensure that a function is defined before its first use, you have to know the order in which statements are executed, which is called the flow of execution.
- Execution always begins at the first statement of the program. Statements are executed one at a time, in order from top to bottom.

18. What is the use of parentheses?

Parentheses have the highest precedence and can be used to force an expression to evaluate in the order you want. It also makes an expression easier to read.

Example: 2 + (3*4) * 7

19. What do you meant by an assignment statement?

An assignment statement creates new variables and gives them values:

```
>>> Message = 'And now for something completely different'
>>> n = 17
```

This example makes two assignments. The first assigns a string to a new variable namedMessage; the second gives the integer 17 to n.

20. What is tuple? (or) What is a tuple? How literals of type tuples are written? Give example(Jan-2018)

A tuple is a sequence of immutable Python objects. Tuples are sequences, like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets. Creating a tuple is as simple as putting different comma-separated values. Comma-separated values between parentheses can also be used.

Example: tup1 = ('physics', 'chemistry', 1997, 2000); tup2= ();

21. Define module.

A module allows to logically organizing the Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that can bind and reference.A module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

22. Name the four types of scalar objects Python has. (Jan-2018)

- int
- float
- bool
- None

23. List down the different types of operator.

Python language supports the following types of operators:

- Arithmetic operator
- Relational operator
- Assignment operator
- Logical operator
- Bitwise operator
- Membership operator
- Identity operator

24. What is a global variable?

Global variables are the one that are defined and declared outside a function and we need to use them inside a function.

Example:#This function uses global variable s

```
def f():
        print(s)
# Global scope
s = "I love India"
f()
```

25. Define function.

A function in Python is defined by a def statement. The general syntax looks like this:

```
def function-name(Parameter list):
        statements # the function body
```


The parameter list consists of zero or more parameters. Parameters are called arguments, if the function is called. The function body consists of indented statements. The function body gets executed every time the function is called. Parameter can be mandatory or optional. The optional parameters (zero or more) must follow the mandatory parameters.

26. What is the purpose of using comment in python program?

Comments indicate information in a program that is meant for other programmers (or anyone reading the source code) and has no effect on the execution of the program. In Python, we use the hash (#) symbol to start writing a comment.

Example:#This is a comment

27. State the reasons to divide programs into functions. (Jan-2019)

Creating a new function gives the opportunity to name a group of statements, which makes program easier to read and debug. Functions can make a program smaller by eliminating repetitive code. Dividing a long program into functions allows to debug the parts one at a time and then assemble them into a working whole. Well designed functions are often useful for many programs.

28. Outline the logic to swap the content of two identifiers without using third variable. (May 2019)

```
a=10
b=20
a=a+b
b=a-b
a=a-b
print("After Swapping a="a," b=",b)
```

29. State about Logical operators available in python language with example. (May 2019)

Logical operators are and, or, not operators.

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

PART B & C (16 MARKS)

1. What is the role of an interpreter? Give a detailed note on python interpreter and interactive mode of operation. (or) Sketch the structures of interpreter and compiler. Detail the differences between them. Explain how Python works in interactive mode and script mode with examples. **(Jan 2019)**
2. Illustrate values and different standard data types with relevant examples.
3. Define variables. List down the rules for naming the variable with example.
4. List down the different types of operators and their function with suitable example.
5. What are the two modes of operation in python? Analyze the differences between them.
6. What do you mean by rule of precedence? List out the order of precedence and demonstrate in detail with example.
7. Elaborate on tuple assignment.
8. What is the use of function? Explain the role of function call and function definition with example. (or) Outline about function definition and call with example. Why are function needed? **(May 2019)**
9. Describe flow of execution of function with an example.
10. Write a Python program to circulate the values of n variables.
11. Write a python program to swap two variables.
12. Write a python program to check whether a given year is a leap year or not.
13. Write a python program to convert celsius to fahrenheit .
    (Formula: celsius * 1.8 = fahrenheit –32).
14. a) What is a numeric literal? Give examples. **(4) (Jan-2018)**
    b) Appraise the arithmetic operators in Python with an example. **(12) (Jan-2018)**
15. a) Outline the operator precedence of arithmetic operators in Python. **(6) (Jan-2018)**
    b) Write a Python program to exchange the value of two variables. **(4) (Jan-2018)**
    c) Write a Python program using function to find a sum of first 'N' even numbers and print the result.**(6)(Jan 2018)**
16. a) Mention the list of keywords available in Python. Compare it with variable name. **(8)(May 2019)**
    b) What is Statement? How are they constructed from variable and expression in Python**. (8) (May 2019)**

UNIT III - CONTROL FLOW, FUNCTIONS, STRINGS
PART- A (2 Marks)

1. **Define Boolean Expression with example.**

A boolean expression is an expression that is either true or false. The values true and false are called boolean values. The following examples use the operator "==" which compares two operands and produces True if they are equal and False otherwise:

Example :>>> 5 == 6

False

True and False are special values that belongs to the type bool; they are not strings:

2. **What are the different types of operators?**

- Arithmetic Operator (+, -, *, /, %, **, //)
- Relational operator (== , !=, <>, < , > , <=, >=)
- Assignment Operator (=, += , *= , - =, /=, %= ,**=)
- Logical Operator (AND, OR, NOT)
- Membership Operator (in, not in)
- Bitwise Operator (& (and), | (or) , ^ (binary Xor), ~(binary 1's complement , << (binary left shift), >> (binary right shift))
- Identity(is, is not)

3. **Explain modulus operator with example.**

The modulus operator works on integers and yields the remainder when the first operand is divided by the second. In Python, the modulus operator is a percent sign (%). The syntax is the same as for other operators:

Example:

```
>>> remainder = 7 % 3
>>> print remainder
1
```

So 7 divided by 3 is 2 with 1 left over.

4. **Explain 'for loop' with example.**

for loops are traditionally used when you have a block of code which you want to repeat a fixed number of times. he Python ***for*** statement iterates over the members of a sequence in order, executing the block each time.

The general form of a for statement is

Syntax:

```
for variable in sequence:
    code block
```

Example:

```
x = 4
for i in range(0, x):
        print i
```

Output:0 1 2 3

5. **Explain relational operators.**

The == operator is one of the relational operators; the others are:

X! = y # x is not equal to y
x > y # x is greater than y
x < y # x is less than y
x >= y # x is greater than or equal to y
x <= y # x is less than or equal to y

6. **Explain while loop with example.(or)Explain flow of execution of while loop with Example.(Jan 2019)**

The statements inside the while loop is executed only if the condition is evaluated to true.

Syntax:

```
while condition:
        statements
```

Example:

```
# Program to add natural numbers upto, sum = 1+2+3+...+10
n = 10
# initialize sum and counter
sum = 0
i = 1
while i <= n:
    sum = sum + i
        i = i+1    # update counter
# print the sum
print("The sum is", sum)
```

7. Explain if-statement and if-else statement with example (or) What are conditional and alternative executions?

If statement:

The simplest form of if statement is:

Syntax:

```
if (condition):
        statement
```

Example:

```
if x > 0:
        print 'x is positive'
```

The boolean expression after 'if' is called the condition. If it is true, then the indented statement gets executed. If not, nothing happens.

If-else:

A second form of if statement is alternative execution, in which there are two possibilities and the condition determines which one gets executed. The syntax looks like this:

Example:

```
if x%2 == 0:
        print 'x is even'
else:
        print 'x is odd'
```

If the remainder when x is divided by 2 is 0, then we know that x is even, and the program displays a message to that effect. If the condition is false, the second set of statements is executed. Since the condition must be true or false, exactly one of the alternatives will be executed.

8. What are chained conditionals?

Sometimes there are more than two possibilities and we need more than two branches. One way to express a computation like that is a chained conditional:

Eg:

```
if x < y:
        print 'x is less than y'
elif x > y:
        print 'x is greater than y'
else:
        print 'x and y are equal'
```

elif is an abbreviation of "else if." Again, exactly one branch will be executed. There is no limit on the number of elif statements. If there is an else clause, it has to be at the end, but there doesn't have to be one.

9. What is a break statement?

When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

Eg:

```
while True:
        line = raw_input('>')
        if line == 'done':
        break
        print line
        print'Done!'
```

10. What is a continue statement?

The continue statement works somewhat like a break statement. Instead of forcing termination, it forces the next iteration of the loop to take place, skipping any code in between.

Example:

```
for num in range(2,10):
        if num%2==0;
                print “Found an even number”, num
                continue
        print “Found a number”, num
```

11. What is recursion? (or) Define Recursion with an example.(Jan 2019) (May 2019)

The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function. Using recursive algorithm, certain problems can be solved quite easily. Examples of such problems are Towers of Hanoi (TOH), Inorder/Preorder/Postorder Tree Traversals, Depth First Search (DFS) of Graph, etc.

Example:

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n-1)
```


12. Compare return value and composition.

Return Value	Composition
Return gives back or replies to the caller of the function. The return statement causes your function to exit and handback a value to its caller. **Example:** def area(radius): temp = math.pi * radius**2 return temp	Calling one function from another is called composition. **Example:** def circle_area(xc, yc, xp, yp): radius = distance(xc, yc, xp, yp) result = area(radius) return result

13. Define string immutability.

Python strings are immutable. ‘a’ is not a string. It is a variable with string value. You can’t mutate the string but can change what value of the variable to a new string.

Program:	**Output:**

a = “foo”
# a now points to foo 	
b=a
# b now points to the same foo that a points to 	
a=a+a
# a points to the new string “foofoo”, but b points to the same old “foo” 	
print a
print b
	#foofoo
	#foo
	It is observed that ‘b’ hasn’t changed even though ‘*a’* has changed.

14. Explain about string module.

The string module contains number of useful constants and classes, as well as some deprecated legacy functions that are also available as methods on strings.

Example:

Program:	**Output:**
import string text = "Monty Python's Flying Circus" print "upper", "=>", string.upper(text) print "lower", "=>", string.lower(text) print "split", "=>", string.split(text) print "join", "=>", string.join(string.split(text), "+") print "replace", "=>", string.replace(text, "Python", "Java") print "find", "=>", string.find(text, "Python"), string.find(text, "Java") print "count", "=>", string.count(text, "n")	upper => MONTY PYTHON'S FLYING CIRCUS lower => monty python's flying circus split => ['Monty', "Python's", 'Flying', 'Circus'] join => Monty+Python's+Flying+Circus replace => Monty Java's Flying Circus find => 6 -1 count => 3

15. Explain global and local scope. (or) Comment with an example on the use of local and global variable with the same identifier name. (May 2019)

The scope of a variable refers to the places that you can see or access a variable. If we define a variable on the top of the script or module, the variable is called global variable. The variables that are defined inside a class or function is called local variable.

Example:

```
def my_local():
        a=10
        print(“This is local variable”)
```

Example:

```
a=10
def my_global():
        print(“This is global variable”)
```

16. Compare string and string slices.

A string is a sequence of character.

Eg: fruit = ‘banana’

String Slices :

A segment of a string is called string slice, selecting a slice is similar to selecting a character.

Eg:>>> s ='Monty Python'

```
>>> print s[0:5]
Monty
>>> print s[6:12]
Python
```

17. Mention a few string functions.

s.captilize() – Capitalizes first character of string

s.count(sub) – Count number of occurrences of sub in string

s.lower() – converts a string to lower case

s.split() – returns a list of words in string

18. What are string methods?

A method is similar to a function. It takes arguments and returns a value. But the syntax is different. For example, the method upper takes a string and returns a new string with all uppercase letters:

Instead of the function syntax upper(word), it uses the method syntax word.upper()

```
.>>> word = 'banana'
>>> new_word = word.upper()
>>> print new_word
BANANA
```

19. Write a Python program to accept two numbers, multiply them and print the result. (Jan-2018)

```
print("Enter two numbers")
val1=int(input())
val2=int(input())
prod=val1*val2
print("The product of the two numbers is:",prod)
```

20. Write a Python program to accept two numbers, find the greatest and print the result. (Jan-2018)

```
print("Enter two numbers")
val1=int(input())
val2=int(input())
if (val1>val2):
        largest=val1
else:
        largest=val2
print("Largest of two numbers is:",largest)
```

21. What is the purpose of pass statement?

Using a pass statement is an explicit way of telling the interpreter to do nothing.

Example:

```
def bar():
        pass
```

If the function bar() is called, it does absolutely nothing.

22. What is range() function?

If you do need to iterate over a sequence of numbers, the built-in function range() comes in handy. It generates arithmetic progressions:

Example:

```
# Prints out the numbers 0,1,2,3,4
for x in range(5):
   print(x)
```

This function does not store all the values in memory, it would be inefficient. So it remembers the start, stop, step size and generates the next number on the go.

23. Define Fruitful Function.

The functions that return values, is called fruitful functions. The first example is area, which returns the area of a circle with the given radius:

```
def area(radius):
temp = 3.14159 * radius**2
return temp
```

In a fruitful function the return statement includes a return value. This statement means: Return immediately from this function and use the following expression as a return value.

24. What is dead code?

Code that appears after a return statement, or any other place the flow of execution can never reach, is called **dead code**.

25. Explain Logical operators

There are three logical operators: and, or, and not. For example, x > 0 and x < 10 is true only if x is greater than 0 and less than 10. n%2 == 0 or n%3 == 0 is true if either of the conditions is true, that is, if the number is divisible

by 2 or 3. Finally, the not operator negates a boolean expression, so not(x > y) is true if x > y is false, that is, if x is less than or equal to y. Non-zero number is said to be true in Boolean expressions.

PART B & C (16 MARKS)

1. Explain in detail various types of operators. **(Jan 2019)**
2. Discuss conditional alternative and chained conditional in detail. **(Jan 2019)**
3. Explain in detail about iterations.
4. Explain in detail about Fruitful Functions.
5. Describe in detail about Recursion.
6. a). Discuss in detail about string slices and string immutability. .**(Jan 2019)** (or) Analyse String slicing. Illustrate how it is done in python with example.**(May 2019)**
   b). Write a python code to search a string in the given list. **(May 2019)**
7. Explain string functions and methods in detail.
8. Write a Python program to find the square root of a number.
9. Explain about string modules in detail.
10. a) Write a Python program to find GCD of two numbers.
    b) Write a Python program to find the exponentiation of a number.
    c) Write a Python program to sum an array of numbers.
11. Explain in detail about list as arrays.
12. a) Write a Python program to perform linear search.
    b) Write a Python program to perform binary search**. (Jan 2019)**
13. a) Appraise with an example nested if and elif header in Python **(6) (Jan 2018)**
    b) Explain with an example while loop, break statement and continue statement in Python. **(10) (Jan 2018)**
14. a) Write a Python program to find the factorial of the given number without recursion with recursion.**(8) (Jan-2018)**
    b) Write a Python program to generate first 'N' Fibonacci series numbers. (Note: Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8… where each number is the sum of the preceding two). **(8) (Jan-2018)**

UNIT IV
LISTS, TUPLES, DICTIONARIES
PART- A (2 Marks)

1. **What is a list?(Jan-2018)**

   A list is an ordered set of values, where each value is identified by an index. The values that make up a list are called its elements. Lists are similar to strings, which are ordered sets of characters, except that the elements of a list can have any type.

2. **Relate String and List? (Jan 2018)(Jan 2019)**

   String:

   String is a sequence of characters and it is represented within double quotes or single quotes. Strings are immutable.

   Example: s=”hello”

   List:

   A list is an ordered set of values, where each value is identified by an index. The values that make up a list are called its elements. Lists are similar to strings, which are ordered sets of characters, except that the elements of a list can have any type and it is mutable.

   Example:

   b= [’a’, ’b’, ’c’, ’d’, 1, 3]

3. **Solve a)[0] * 4 and b) [1, 2, 3] * 3.**

```
>>> [0] * 4
[0, 0, 0, 0]
>>> [1, 2, 3] * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

4. **Let list = [’a’, ’b’, ’c’, ’d’, ’e’, ’f’]. Find a) list[1:3] b) t[:4] c) t[3:] .**

```
>>> list = [’a’, ’b’, ’c’, ’d’, ’e’, ’f’]
>>> list[1:3]
[’b’, ’c’]
>>> list[:4]
[’a’, ’b’, ’c’, ’d’]
>>> list[3:]
[’d’, ’e’, ’f’]
```


5. **Mention any 5 list methods.**
   - append()
   - extend ()
   - sort()
   - pop()
   - index()
   - insert
   - remove()

6. **State the difference between lists and dictionary.**

Lists	Dictionary
• List is a mutable type meaning that it can be modified. • List can store a sequence of objects in a certain order. • **Example:** list1=[1,’a’,’apple’]	• Dictionary is immutable and is a key value store. • Dictionary is not ordered and it requires that the keys are hashtable. • **Example:** dict1={‘a’:1, ‘b’:2}

7. **What is List mutability in Python? Give an example.**

Python represents all its data as objects. Some of these objects like **lists** and dictionaries are **mutable**, i.e., their content can be changed without changing their identity. Other objects like integers, floats, strings and tuples are objects that cannot be changed.

Example:

```
>>> numbers = [17, 123]
>>> numbers[1] = 5
>>> print numbers [17, 5]
```

8. What is aliasing in list? Give an example.

An object with more than one reference has more than one name, then the object is said to be aliased.

Example: If *a* refers to an object and we assign b = a, then both variables refer to the same object:

```
>>> a = [1, 2, 3]
>>> b = a
>>> b is a True
```

9. Define cloning in list.

In order to modify a list and also keep a copy of the original, it is required to make a copy of the list itself, not just the reference. This process is sometimes called cloning, to avoid the ambiguity of the word "copy".

Example:

```
def Cloning(li1):
    li_copy = li1[:]
    return li_copy
# Driver Code
li1 = [4, 8, 2, 10, 15, 18]
li2 = Cloning(li1)
print("Original List:", li1)
print("After Cloning:", li2)
```

Output:

```
Original List: [4, 8, 2, 10, 15, 18]
After Cloning: [4, 8, 2, 10, 15, 18]
```


10. Explain List parameters with an example.

Passing a list as an argument actually passes a reference to the list, not a copy of the list. For example, the function head takes a list as an argument and returns the first element:

Example:

```
def head(list):
        return list[0]
```

Here's how it is used:

```
>>> numbers = [1, 2, 3]
>>> head(numbers)
>>> 1
```

11. Write a program in Python returns a list that contains all but the first element of the given list.

```
def tail(list):
return list[1:]
```

Here's how tail is used:

```
>>> numbers = [1, 2, 3]
>>> rest = tail(numbers)
>>> print rest [2, 3]
```

12. Write a program in Python to delete first element from a list.

```
def deleteHead(list): del list[0]
```

Here's how deleteHead is used:

```
>>> numbers = [1, 2, 3]
>>> deleteHead(numbers)
>>> print numbers [2, 3]
```

13. What is the benefit of using tuple assignment in Python?

It is often useful to swap the values of two variables. With conventional assignments a temporary variable would be used.

For example, to swap a and b:

```
>>> temp = a
>>> a = b
>>> b = temp
```

This solution is cumbersome; tuple assignment is more elegant:

```
>>> a, b = b, a
```

14. Define key-value pairs.

The elements of a dictionary appear in a comma-separated list. Each entry contains an index and a value separated by a colon. In a dictionary, the indices are called keys, so the elements are called key-value pairs.

15. Define dictionary with an example.

A dictionary is an associative array (also known as hashes). Any key of the dictionary is associated (or mapped) to a value. The values of a dictionary can be any Python data type. So dictionaries are unordered key-value-pairs.

Example:

```
>>> eng2sp = {}  # empty dictionary
>>> eng2sp['one'] = 'uno'
>>> eng2sp['two'] = 'dos'
```

16. How to return tuples as values?

A function can only return one value, but if the value is a tuple, the effect is the same as returning multiple values. For example, if you want to divide two integers and compute the quotient and remainder, it is inefficient to compute x/y and then x%y. It is better to compute them both at the same time.

```
>>> t = divmod(7, 3)
>>> print t (2, 1)
```

17. List two dictionary operations.

- Del -removes key-value pairs from a dictionary
- Len - returns the number of key-value pairs

18. Define dictionary methods with an example.

A method is similar to a function. It takes arguments and returns a value but the syntax is different. For example, the keys method takes a dictionary and returns a list of the keys that appear, but instead of the **function syntax** keys (dictionary_name), **method syntax** dictionary_name.keys() is used.

Example: >>> eng2sp.keys() ['one', 'three', 'two']

19. Define List Comprehension.

List comprehensions apply an arbitrary expression to items in an iterable rather than applying function. It provides a compact way of mapping a list into another list by applying a function to each of the elements of the list.

20. Write a Python program to swap two variables.

```
x = 5
y = 10
temp = x
x = y
y = temp
print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

21. Write the syntax for list comprehension.

The list comprehension starts with a '**[' and ']'**, to help you remember that the result is going to be a list. The basic syntax is[expression for item in list if conditional].

Example:

```
new_list = []
for i in old_list:
   if filter(i):
      new_list.append(expressions(i))
```

22. How list differs from tuple. (Jan-2018)

List	Tuple
• List is a mutable type meaning that it can be modified. • **Syntax:** list=[]	• Tuple is an immutable type meaning that it cannot be modified. • **Syntax:** tuple=()

• **Example:** list1=[1,'a']	• **Example:** tuple1=(1,'a')

23. How to slice a list in Python. (Jan-2018)

The values stored in a list can be accessed using slicing operator, the colon (:) with indexes starting at 0 in the beginning of the list and end with -1.

Example:

```
>>>  list = ['a', 'b', 'c', 'd', 'e', 'f']
>>> list[1:3]['b',
'c']
```

24. Write python program for swapping two numbers using tuple assignment?

```
a=10
b=20
a,b=b,a
print("After swapping a=%d,b=%d"%(a,b))
```

25. What is list loop?

In Python lists are considered a type of iterable . An iterable is a data type that can return its elements separately, i.e., one at a time.

Syntax:

```
for <item> in <iterable>:
    <body>
```

Example:

```
>>>names = ["Uma","Utta","Ursula","Eunice","Unix"]
>>>for name in names:
        print("Hi "+ name +"!")
```

26. What is mapping?

A list is a relationship between indices and elements. This relationship is called a **mapping**; each index "maps to" one of the elements. The **in** operator also works on lists.

```
>>> cheeses = ['Cheddar', 'Edam', 'Gouda']
>>> 'Edam' in cheeses
True
>>> 'Brie' in cheeses
False
```


27. Give a function that can take a value and return the first key mapping to that value in a dictionary. (Jan 2019)

```
a={'aa':2, "bb":4}
print(a.keys()[0])
```

28. How to create a list in python? Illustrate the use of negative indexing of list with example. (May 2019)

List Creation:

```
days = ['mon', 2]
days=[]
days[0]='mon'
days[1]=2
```

Negative Indexing:

Example:

```
>>> print(days[-1])
```

Output: 2

29. Demonstrate with simple code to draw the histogram in python. (May 2019)

```
def histogram( items ):
    for n in items:
```

```
        output = ''
        times = n
        while( times > 0 ):
          output += '*'
        times = times - 1
        print(output)
histogram([2, 3, 6, 5])
```

Output:

```
**
***
******
*****
```

PART B & C (16 MARKS)

1. Explain in detail about lists, list operations and list slices. **(Jan-2019)**
2. Discuss in detail about list methods and list loops with examples.
3. Explain in detail about mutability and tuples with a Python program.
4. What is tuple assignment? Explain it with an example.
5. Is it possible to return tuple as values? Justify your answer with an example.
6. Explain in detail about dictionaries and its operations.(or)What is a dictionary in Python?Give example.**(4) (Jan-2018)**
7. Describe in detail about dictionary methods.(or) What is Dictionary? Give an example. **(4) (May 2019)**
8. Explain in detail about list comprehension .Give an example.
9. Write a Python program for
   a) selection sort **(8) (Jan-2018)**
   b) Insertion sort.

10. Write a Python program for
    a) Merge sort **(May 2019)**
    b) Quick sort.
11. Appraise the operations for dynamically manipulating dictionaries **(12) (Jan-2018)**
12. Write a Python program to perform linear search on a list **(8) (Jan-2018)**
13. Demonstrate with code the various operations that can be performed on tuples. **(May 2019)**

UNIT V
FILES, MODULES, PACKAGES
PART- A (2 Marks)

1. **What is a text file?**
   A text file is a file that contains printable characters and whitespace, organized in to lines separated by newline characters.
2. **Write a python program that writes "Hello world" into a file.**
```
f =open("ex88.txt",'w')
f.write("hello world")
f.close()
```
3. **Write a python program that counts the number of words in a file.**
```
f=open("test.txt","r")
content =f.readline(20)
words =content.split()
print(words)
```
4. **What are the two arguments taken by the open() function?**
   The open function takes two arguments : name of the file and the mode of operation.
   Example: f = open("test.dat","w")
5. **What is a file object?**
   A file object allows us to use, access and manipulate all the user accessible files. It maintains the state about the file it has opened.
   Example: f = open("test.dat","w") // f is the file object.
6. **What information is displayed if we print a file object in the given program?**
```
 f= open("test.txt","w")
 print f
```
   The name of the file, mode and the location of the object will be displayed.
7. **What is an exception?**
   Whenever a runtime error occurs, it creates an exception. The program stops execution and prints an error message.
   Example:
```
 #Dividing by zero creates an exception:
            print 55/0
ZeroDivisionError: integer division or modulo
```
8. **What are the two parts in an error message?**
   The error message has two parts: the type of error before the colon, and specification about the error after the colon.
   Example:
```
>>> 10 * (1/0)
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ZeroDivisionError: integer division or modulo by zero
```
9. **What are the error messages that are displayed for the following exceptions?**
   a. Accessing a non-existent list item
   b. Accessing a key that isn't in the dictionary
   c. Trying to open a non-existent file
   a. IndexError: list index out of range

b. KeyError: what
c. IOError: [Errno 2] No such file or directory: 'filename'

10. How do you handle the exception inside a program when you try to open a non-existent file?

```
filename = raw_input('Enter a file name: ')
try:
f = open (filename, "r")
except IOError:
print 'There is no file named', filename
```

11. How does try and execute work?

The *try* statement executes the statements in the first block. If no exception occurs, then *except* statement is ignored. If an exception of type *IOError* occurs, it executes the statements in the *except branch* and then continues.

Syntax:

```
try:
  // try block code
except:
  // except blcok code
```

Example:

```
try:
   print "Hello World"
except:
   print "This is an error message!"
```


12. What is the function of raise statement? What are its two arguments?

The raise statement is used to raise an exception when the program detects an error. It takes two arguments: the exception type and specific information about the error.

13. What is a pickle?

Pickling saves an object to a file for later retrieval. The pickle module helps to translate almost any type of object to a string suitable for storage in a database and then translate the strings back in to objects.

14. What is the use of the format operator?

The format operator % takes a format string and a tuple of expressions and yields a string that includes the expressions, formatted according to the format string.

Example:

```
>>> nBananas = 27
>>> "We have %d bananas." % nBananas
'We have 27 bananas.'
```

15. What are the two methods used in pickling?

The two methods used in pickling are

1. pickle.dump()

2. pickle.load().

To store a data structure, dump method is used and to load the data structures that are dumped, load method is used.

16. What are modules?(or) Write a note on modular design (Jan-2018)

- Modules are files containing Python definitions and statements (ex: name.py)
- Modules can contain executable statements along with function definitions.
- Each modules has its own private symbol table used as the global symbol table all functions in the module.
- Modules can import other modules.

Syntax:

```
import <modulename>
```

17. What is a package?

Packages are namespaces that contain multiple packages and modules themselves. They are simply directories.

Syntax:

```
from <mypackage> import <modulename>
```

Example: from Game.Level.start import select_difficulty

18. Write a Python script to display the current date and time. (Jan-2018)

```
import datetime
print("date and time", datetime.datetime.now())
```

19. What is the special file that each package in Python must contain?

Each package in Python must contain a special file called __init_.py. __init_.py can be an empty file but it is often used to perform setup needed for the package(import things, load things into path, etc).

Example :


```
package/
   __init_.py
   file.py
   file2.py
   file3.py
   subpackage/
      __init_.py
      submodule1.py
      submodule2.py
```

20. How do you delete a file in Python?

The remove() method is used to delete the files by supplying the name of the file to be deleted as argument.

Syntax:

```
os.remove(filename)
```

Example:

```
import os
os.remove("ChangedFile.csv")
print("File Removed!")
```

21. How do you use command line arguments to give input to the program? (or) What is command line argument? (May 2019)

Python sys module provides access to any command-line arguments via sys.argv. sys.argv is the list of command-line arguments. len(sys.argv) is the number of command-line arguments.

Example:

```
import sys
program_name = sys.argv[0]
arguments = sys.argv[1:]
count = len(arguments)
```

22. What are the different file operations?

In Python, a file operation takes place in the following order.

1. Open a file
2. Read or write (perform operation)
3. Close the file

23. What are the different file modes?

- 'r' - Open a file for reading. (default)
- 'w - Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
- 'x' - Open a file for exclusive creation. If the file already exists, the operation fails.
- 'a' - Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
- 't' - Open in text mode. (default)
- 'b' - Open in binary mode.
- '+' - Open a file for updating (reading and writing)

24. How to view all the built-in exception in python.

The built-in exceptions using the `local()` built-in functions as follows.

Syntax:

```
>>> locals()['__builtins__']
```


This will return us a dictionary of built-in exceptions, functions and attributes.

25. What do you mean IndexError?

IndexError is raised when index of a sequence is out of range.

Example:

```
>>> l=[1,2,3,4,5]
>>> print l[6]
Traceback (most recent call last):
File "<pyshell#16>", line 1, in <module>
print l[6]
IndexError: list index out of range
```

26. Find the syntax error in the code given:

while True print('Hello World') (Jan 2019)

In the above given program, colon is missing after the condition. The right way to write the above program is

```
while True:
        print('Hello World')
```

27. Categorize the different types errors arises during programming. Interpret the following python code (May 2019)

```
>>>import os
>>>cwd = os.getcwd()
>>>print cwd
```

Basic types of errors:

Syntax Error:

Raised by the parser when a syntax error is encountered.

Semantic Error:

Raised by the parser when there is logical error in the program.

Here in the above given program, Syntax error occurs in the third line (**print cwd**)

SyntaxError: Missing parentheses in call to 'print'.

PART B & C (16 MARKS)

1. Write a function that copies a file reading and writing up to 50 characters at a time. (or) Explain the commands used to read and write into a file with example. **(Jan 2019)**
2. (a). Write a program to perform exception handling.
   (b). Write a Python program to handle multiple exceptions.
3. Write a python program to count number of lines, words and characters in a text file. **(May 2019)**
4. Write a Python program to illustrate the use of command-line arguments.
5. Mention the commands and their syntax for the following: get current directory, changing directory, list, directories and files, make a new directory, renaming and removing directory.
6. Write a Python program to implement stack operations using modules.
7. Write a program to illustrate multiple modules.
8. Write a Python program to dump objects to a file using pickle.
9. Tabulate the different modes for opening a file and briefly explain the same. **(Jan-2018) (Jan 2019)**
10. (a). Appraise the use of try block and except block in Python with example. **(6) (Jan-2018)**
    (b). Explain with example exceptions with argument in Python. **(10) (Jan-2018)**
11. a) Describe how exceptions are handled in python with necessary examples.(8) **(Jan 2019, May 2019)**
    b) Discuss about the use of format operator in file processing.(8) (or) Explain about the file reading and writing operations using format operator with python code.(16) **(Jan 2019, May 2019)**