



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu.



MUST KNOW CONCEPTS

MKC

MCA

2021-22

Course Code & Course Name : 21CAB04 & Problem Solving and Python Programming

Year/Sem/Sec : I / I / -

S.No.	Term	Notation (Symbol)	Concept / Definition / Meaning / Units / Equation / Expression	Units
Unit-I : Introduction To Computing And Problem Solving				
1.	SWOT	--	Strengths, Weaknesses, Opportunities, Threats	I
2.	IDEAL	--	Identify, Define, Explore, Action, Look back.	I
3.	PEST	--	Political, Economic, Social, Technological	I
4.	SLEPT	--	Social, Legal, Economic, Political, Technological.	I
5.	PMI	--	A decision-making strategy created by Edward de Bono. For any problem or solution, list these: Plus Points, Minus Points, Interesting Points	I
6.	FFOE	--	A creativity technique: Fluency (many ideas), Flexibility (variety of ideas), Originality (unique ideas), Elaboration (fully developed ideas).	I
7.	DO IT	--	A simple process for creativity: Define problem, open mind and apply creative techniques, identify best solution and transform	I
8.	PCD	--	Possibilities, Consequences, Decision	I
9.	GRASP	--	Getting Results And Solving Problems	I
10.	PACRA	--	Purpose, Alternatives, Criteria, Resources, Action	I
11.	S.O.D.A.S.	--	S=Situation, O=Options, D=Disadvantages, A=Advantages, S = Solution	I
12.	CAP	--	Cover All Possibilities.	I
13.	Value	--	A value is one of the fundamental things — like a letter or a number — that a program manipulates.	I
14.	Variables	--	A variable is a name that refers to a value	I
15.	Keywords	--	Keywords define the language's rules and structure, and they cannot be used as variable names	I
16.	Statement	--	A statement is an instruction that the Python interpreter can execute.	I

17.	Expression	--	An expression is a combination of values, variables, and operators.	I
18.	Operators	--	Operators are special symbols that represent computations like addition and multiplication.	I
19.	Operands	--	The values the operator uses are called operands	I
20.	Rules of precedence	--	When more than one operator appears in an expression, the order of evaluation depends on the rules of precedence.	I
21.	Comments	--	It is a good idea to add notes to your programs to explain in natural language what the program is doing.	I
22.	Concatenate	--	To join two strings end-to-end.	I
23.	Pseudocode	--	Pseudocode is an artificial and informal language that helps programmers develop algorithms.	I
24.	Flowchart	--	A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes.	I
25.	State diagram	--	A graphical representation of a set of variables and the values to which they refer.	I
Unit-II : Conditionals And Functions				
26.	Program	--	A program is a sequence of instructions that specifies how to perform a computation.	II
27.	Input	--	Get data from the keyboard, a file, or some other device.	II
28.	Output	--	Display data on the screen or send data to a file or other device.	II
29.	Math	--	Perform basic mathematical operations like addition and multiplication.	II
30.	Conditional execution	--	Check for certain conditions and execute the appropriate code.	II
31.	Repetition	--	Perform some action repeatedly, usually with some variation.	II
32.	Debugging	--	Programming errors are called bugs and the process of tracking them down is called debugging	II
33.	Syntax	--	Syntax refers to the structure of a program and the rules about that structure.	II
34.	Natural languages	--	Natural languages are the languages people speak, such as English, Spanish, and French	II
35.	Formal languages	--	Formal languages are languages that are designed by people for specific applications.	II
36.	Problem solving	--	The process of formulating a problem, finding a solution, and expressing the solution.	II
37.	High-level language	--	A programming language like Python that is designed to be easy for humans to read and write	II

38.	Low-level language	--	A programming language that is designed to be easy for a computer to execute; also called “machine language” or “assembly language.	II
39.	Portability	--	A property of a program that can run on more than one kind of computer.	II
40.	Interpret	--	To execute a program in a high-level language by translating it one line at a time	II
41.	Compile	--	To translate a program written in a high-level language into a low-level language all at once, in preparation for later execution.	II
42.	Source code	--	A program in a high-level language before being compiled.	II
43.	Object code	--	The output of the compiler after it translates the program.	II
44.	Prompt	--	Characters displayed by the interpreter to indicate that it is ready to take input from the us.	II
45.	Script:	--	A program stored in a file (usually one that will be interpreted).	II
46.	Interactive mode	--	A way of using the Python interpreter by typing commands and expressions at the prompt	II
47.	Script mode	--	A way of using the Python interpreter to read and execute statements in a script.	II
48.	Algorithm	--	A general process for solving a category of problems	II
49.	Syntax error	--	An error in a program that makes it impossible to parse.	II
50.	Exception	--	An error that is detected while the program is running.	II
Unit-III : Simple Data Structures In Python				
51.	PEMDAS	--	Parentheses, Exponentiation, Multiplication and Division, Addition and Subtraction,	III
52.	Function	--	A function is a named sequence of statements that performs a computation.	III
53.	Module	--	A module is a file that contains a collection of related functions	III
54.	Function definition	--	A function definition specifies the name of a new function and the sequence of statements that execute when the function is called.	III
55.	Header	--	The first line of the function definition is called the header	III
56.	Body	--	The first line of the function definition is called the header; the rest is called the body.	III
57.	Flow of execution	--	the order in which statements are executed, which is called the flow of execution	III
58.	Fruitful functions	--	Some of the functions we are using, such as the math functions, yield results; for lack of a better name, I call them fruitful functions.	III
59.	Void functions	--	Other functions, like print_twice, perform an action but don't return a value. They are called void functions.	III

60.	Parameter	--	A name used inside a function to refer to the value passed as an argument	III
61.	Function call	--	A statement that executes a function. It consists of the function name followed by an argument list.	III
62.	Argument	--	A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function	III
63.	Local variable	--	A variable defined inside a function. A local variable can only be used inside its function	III
64.	Return value	--	The result of a function. If a function call is used as an expression, the return value is the value of the expression	III
65.	Import statement	--	A statement that reads a module file and creates a module object	III
66.	Frame	--	A box in a stack diagram that represents a function call. It contains the local variables and parameters of the function.	III
67.	A Boolean expression	--	A Boolean expression is an expression that is either true or false.	III
68.	Chained conditional	--	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	III
69.	Recursion	--	It is legal for one function to call another; it is also legal for a function to call itself	III
70.	String	--	A string is a sequence of characters.	III
71.	String slices	--	A segment of a string is called a slice	III
72.	List	--	The list is a data type available in Python which can be written as a list of comma-separated values between square brackets.	III
73.	Tuple	--	Tuple is a collection of Python objects much like a list. The sequence of values stored in a tuple can be of any type, and they are indexed by integers.	III
74.	Set	--	Set is an unordered collection of data type that is iterable, mutable, and has no duplicate elements.	III
75.	Characteristics of Python lists	--	Lists are ordered, contain any arbitrary objects, mutable, dynamic.	III
Unit-IV: Strings, Dictionaries, Modules				
76.	Accessing Values in Strings	--	To access substrings, use the square brackets for slicing along with the index or indices to obtain your substring	IV
77.	Updating Strings	--	You can "update" an existing string by (re)assigning a variable to another string.	IV
78.	Triple Quotes	--	Python's triple quotes comes to the rescue by allowing strings to span multiple lines,	IV

			including verbatim NEWLINES, TABs, and any other special characters.	
79.	r'expression'	--	print r'C:\\nowhere' When the above code is executed, it produces the following result – C:\\nowhere	IV
80.	Unicode String	--	print u'Hello, world!' When the above code is executed, it produces the following result – Hello, world!	IV
81.	String Membership Test	--	>>> 'a' in 'program' True >>> 'at' not in 'battle' False	IV
82.	String Slicing	--	To access a range of characters in the String, method of slicing is used. Slicing in a String is done by using a Slicing operator (colon).	IV
83.	Deleting Entire String	--	Deletion of entire string is possible with the use of del keyword. del String1	IV
84.	Dictionary	--	Dictionaries are used to store data values in key:value pairs. thisdict={ "brand": "Ford", "model": "Mustang", "year": 1964 }	IV
85.	Dictionary items	--	Dictionary items are presented in key: value pairs, and can be referred to by using the key name.	IV
86.	Dictionary Length	--	To determine how many items a dictionary has, use the len() function:	IV
87.	<u>keys()</u>	--	Returns a list containing the dictionary's keys x = car.keys()	IV
88.	<u>items()</u>	--	Returns a list containing a tuple for each key value pair x = car.items()	IV
89.	<u>values()</u>	--	Returns a list of all the values in the dictionary x = car.values()	IV
90.	<u>clear()</u>	--	Removes all the elements from the dictionary car.clear()	IV
91.	<u>get()</u>	--	Returns the value of the specified key x = car.get("model")	IV
92.	<u>pop()</u>	--	Removes the element with the specified key car.pop("model")	IV
93.	<u>copy()</u>	--	Returns a copy of the dictionary x = car.copy()	IV
94.	Module	--	A module is a file containing Python code. A package, however, is like a directory that holds sub-packages and modules.	IV
95.	Package	--	A package must hold the file __init__.py. This does not apply to modules.	IV
96.	Locating Modules	--	The current directory, If the module isn't found, Python then searches each directory in the shell variable PYTHONPATH, If all else fails, Python checks the default path. On UNIX, this default path is normally /usr/local/lib/python/.	IV
97.	PYTHONPATH	--	An environment variable, consisting of a list of directories	IV
98.	namespace	--	A namespace is a dictionary of variable names	IV

			(keys) and their corresponding objects (values).	
99.	dir()	--	The dir() built-in function returns a sorted list of strings containing the names defined by a module.	IV
100.	globals() and locals()	--	The globals() and locals() functions can be used to return the names in the global and local namespaces depending on the location from where they are called.	IV
Unit-V : File Handling And Exception Handling				
101.	Open a File	--	f = open("demofile.txt", "r") print(f.read())	V
102.	Read Only Parts of the File	--	f = open("demofile.txt", "r") print(f.read(5))	V
103.	Read Lines	--	f = open("demofile.txt", "r") print(f.readline())	V
104.	Close File	--	f = open("demofile.txt", "r") print(f.readline()) f.close()	V
105.	Write to an Existing File	--	To write to an existing file, you must add a parameter to the open() function: "a" - Append - will append to the end of the file "w" - Write - will overwrite any existing content	V
106.	Create a New File	--	To create a new file in Python, use the open() method, with one of the following parameters : "x" - Create, "a" - Append, "w" - Write	V
107.	Delete a File	--	To delete a file, you must import the OS module, and run its os.remove() function:	V
108.	Check if File exist:	--	import os if os.path.exists("demofile.txt"): os.remove("demofile.txt")else: print("The file does not exist")	V
109.	Delete Folder	--	To delete an entire folder, use the os.rmdir() method	V
110.	Two file modes	--	"t" - Text - Default value. Text mode and "b" - Binary - Binary mode (e.g. images)	V
111.	Working of append() mode	--	file = open('geek.txt','a') file.write("This will add this line") file.close()	V
112.	Creating a file using write() mode	--	file = open('geek.txt','w') file.write("This is the write command") file.close()	V
113.	split() using file handling	--	with open("file.txt", "r") as file:data = file.readlines() for line in data: word = line.split() print (word)	V
114.	Files	--	Files are used to permanently store data in a non-volatile memory (e.g. hard disk).	V
115.	Order in file operation	--	1. Open a file 2. Read or write (perform operation) 3. Close the file	V

116.	f.tell() f.seek(0)	--	get the current file position bring file cursor to initial position	V
117.	tell()	--	Returns the current file location.	V
118.	Exceptions	--	Errors detected during execution are called exceptions	V
119.	Zero Division Error	--	>>> 10 * (1/0) Traceback (most recent call last): File "<stdin>", line 1, in <module> ZeroDivisionError: division by zero	V
120.	Name Error	--	>>>4 + spam*3 Traceback (most recent call last): File "<stdin>", line 1, in <module> NameError: name 'spam' is not defined	V
121.	Type Error	--	>>>'2' + 2 Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: Can't convert 'int' object to str implicitly	V
122.	Catching Exceptions in Python	--	In Python, exceptions can be handled using a try statement.	V
123.	try .. except	--	A try clause can have any number of except clauses to handle different exceptions, however, only one will be executed in case an exception occurs.	V
124.	Raising Exceptions in Python	--	In Python programming, exceptions are raised when errors occur at runtime. We can also manually raise exceptions using the raise keyword.	V
125.	Python try...finally	--	try: f = open("test.txt",encoding = 'utf-8') # perform file operations finally: fclose()	V

Placement Questions

126.	What type of language is python?	--	Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.	
127.	Python an interpreted language. Explain.	--	An interpreted language is any programming language which is not in machine-level code before runtime. Therefore, Python is an interpreted language.	
128.	What is pep 8?	--	PEP stands for Python Enhancement Proposal. It is a set of rules that specify how to format Python code for maximum readability.	
129.	What is namespace in Python?	--	A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.	
130.	PYTHONPATH	--	It is an environment variable which the user can set to add additional directories that the user	

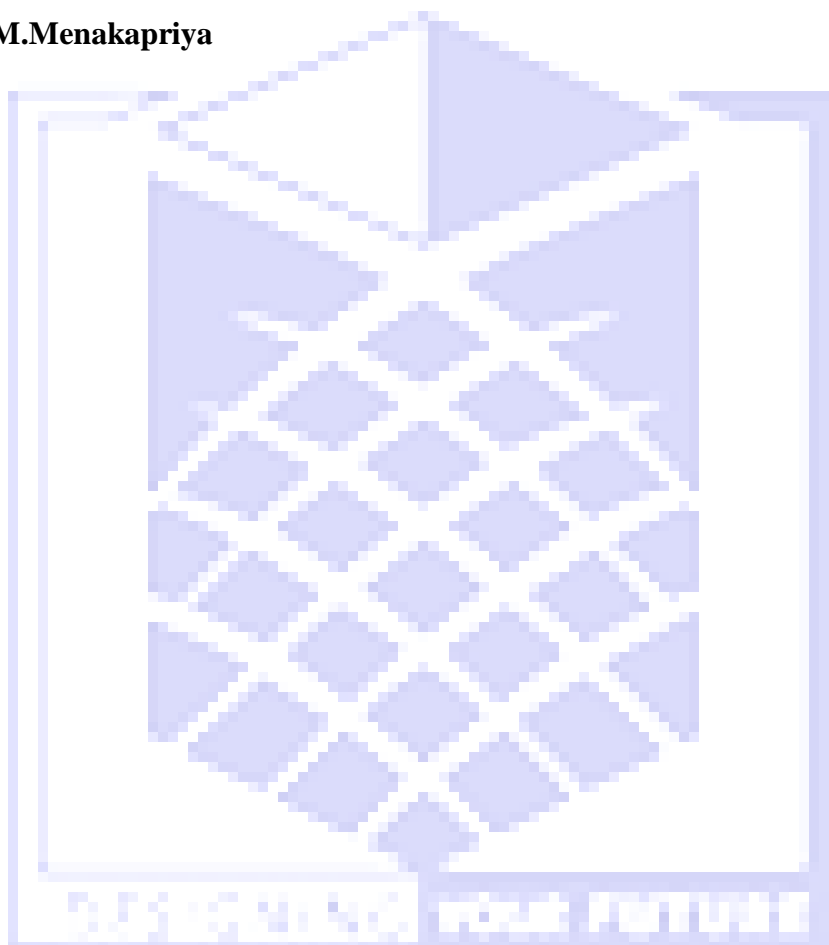
			wants Python to add to the sys.	
131.	What are python modules?	--	Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.	
132.	What are local variables in Python?	--	Any variable declared inside a function is known as a local variable.	
133.	What are global variables in Python?	--	Variables declared outside a function or in global space are called global variables.	
134.	Is python case sensitive?	--	Yes. Python is a case sensitive language.	
135.	What is type conversion in Python?	--	Type conversion refers to the conversion of one data type into another.	
136.	Is indentation required in python?	--	Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block.	
137.	What is the difference between Python Arrays and lists?	--	Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type element whereas lists can hold any data type elements.	
138.	What are functions in Python?	--	A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used.	
139.	What is __init__?	--	This method called when an object is created from the class and it allow the class to initialize the attributes of a class.	
140.	What is a lambda function?	--	An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.	
141.	What is self in Python?	--	Self is an instance or an object of a class. In Python, this is explicitly included as the first parameter.	
142.	How does break work?	--	Allows loop termination when some condition is met and the control is transferred to the next statement.	
143.	How does continue work?	--	Allows skipping some part of a loop when some specific condition is met and the control is transferred to the beginning of the loop	
144.	How does pass work?	--	Used when you need some block of code syntactically, but you want to skip its execution. This is basically a null operation. Nothing happens when this is executed.	
145.	What does [::-1] do?	--	[::-1] is used to reverse the order of an array or a sequence.	
146.	How can you randomize the items of a list in place in Python?	--	From random import shuffle: x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High'] shuffle(x), print(x) O/P: ['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']	
147.	What are python iterators?	--	Iterators are objects which can be traversed though or iterated upon.	

148.	What are docstrings in Python?	--	Python docstrings are the string literals that appear right after the definition of a function, method, class, or module.	
149.	Pickling	--	It is used for implementing binary protocols for serializing and de-serializing a Python object structure.	
150.	Unpickling	--	It is the process of retrieving original python objects from the stored string representation	

Faculty Prepared

Signature

Mrs.M.Menakapriya



HoD

Estd. 2000