



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

Value Added Course Summary 2018-2019

Course Name: Solving Problems using Python Programming

Course Duration: 32 hours

Year Offered: 2018-2019

Course Instructors: Ms. V. Nivethitha, Assistant Professor/CSE

Ms. S. Bharathi, Assistant Professor/CSE

Ms. N. Padma Priya, Assistant Professor/CSE

Course Outcome:

On completion of the course, students will be able to Develop algorithmic solutions to simple computational problems and to develop and execute simple Python programs.

Course Type: Self Framed

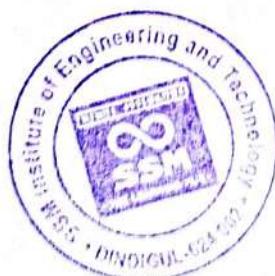
Assessment Mode

Attendance: 32 hours

Number of Participants: 35

Scheme of Exam: Python programs as Assignment

✓ Sent
Course Co-ordinator



D.Senthil Kumaran
Dr.D.SENTHIL KUMARAN, M.E., Ph.D., (NUS)
Principal
SSM Institute of Engineering and Technology
Kuttathupatti Village, Sindagundu (Po),
Palani Road, Dindigul - 624 002

V.Jayaram
V.JAYARAM
HoD

24.07.2018

From,

Ms. S. Bharathi, Ms. V. Nivedhitha, Ms. N. Padma Priya,
Assistant Professors, Department of CSE,
SSM Institute of Engineering and Technology,
Dindigul.

V-19
III-XIV CSE

Through,

The Head of the Department,
Department of CSE,
SSM Institute of Engineering and Technology,
Dindigul.

To,

The Principal,
SSM Institute of Engineering and Technology,
Dindigul.

Respected Sir,

Sub: Requisition for conducting Value added course for CSE students - Reg.

This is to let you know that we have planned to conduct a Value Added Course on
“Solving Problems using Python Programming” from 06.08.2018 onwards for our III and
IV CSE students. In this regard, we kindly request you to provide permission for the
same.

The course details are attached herewith for your reference.

Thank You.

1. *S.Bharathi* [S. BHARATHI]
2. *J.V.Nivedhitha* [V. NIVEDHITHA]
3. *N.Padmapriya* [N. Padmapriya]
Faculty In-charge

V. J. MUTHU
HOD/CSE

(Dr. V. SHANMUGHAMAL)

T. Ramya
PRINCIPAL



Dr. D. SENTHIL KUMARAN M.E., Ph.D.
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering and Technology
Palani Road, Dindigul-624 002

Technology Training

Solving Problem Using Python Programming

Instructurs

V. Nivedhitha

S. Bharathi

N. Padmapriya

Let's
Join
Us!



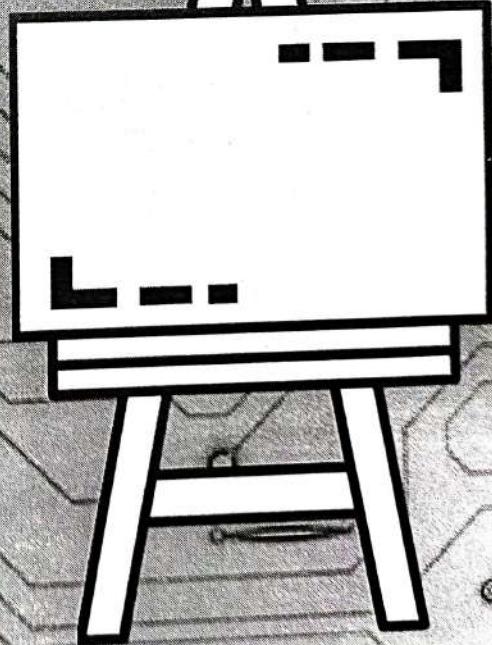
HOD/CSE

DR. V. SHANMUGAVEL

Dr.D.GENTH KUMARAN M.E.I.E.T.
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Tirunindigul - 624 002

PRINCIPAL

M. V. SARAVANAN



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul - Palani Highway, Dindigul - 624 002.



Department of Computer Science and Engineering

**Value added course
on**

"Solving Problems using Python programming"

Course Schedule

Course Venue: Computer lab - 2 (Hands on training)

Lecture Hall: A-202

Timing: 4.30 p.m. to 5.30 p.m.

Sl. No.	Date	Topic to be covered	Course Instructor
1.	06.08.2018	Python interpreter and interactive mode, Values and types: int, float, boolean	N. Padma Priya, V. Nivedhitha
2.	07.08.2018	Operators, Precedence of operators	
3.	08.08.2018	Conditionals: if, if-else, if-elif-else	
4.	09.08.2018	Iteration: while, for, break, continue, pass	
5.	10.08.2018	Strings, string slices, String methods	
6.	16.08.2018	Lists, list slicing, List methods, cloning and aliasing lists	V. Nivedhitha, S. Bharathi
7.	17.08.2018	Tuples, Tuple assignment	
8.	18.08.2018	Dictionaries, operations and methods	
9.	20.08.2018	Modules and functions, function definition and use	
10.	21.08.2018	Parameters and arguments	
11.	23.08.2018	Fruitful and Recursive functions	S. Bharathi, N. Padma Priya
12.	24.08.2018	Files, file operations	
13.	25.08.2018	Errors and Exceptions, Handling exceptions	
14.	27.08.2018	Command Line Arguments	
15.	28.08.2018	Modules and Packages	
16.	29.08.2018	Simple Applications	Ms. S. Bharathi, Ms. V. Nivedhitha, Ms. N. Padma Priya
17.	30.08.2018	Simple Applications	
18.	31.08.2018	Project Idea Presentation(Proposal Submission)	
19.	12.09.2018	Submission of Mini-Project Report	
20.	15.09.2018	Mini-Project Competition	

1. *S. Bharathi [S. Bharathi]*
2. *V. Nivedhitha [V. Nivedhitha]*
3. *N. Padma Priya [N. Padma Priya]*

Faculty In-charge

Dr.D.SENTHIL KUMARAN M.E.,F.O
 Senior Professor
 Department of Mechanical Engineering
 SSM Institute of Engineering Technology
 Palani Road, Dindigul-624 002



*V. Nivedhitha
24/07/18*
HOD/CSE



*Dinesh
24/07/18*
PRINCIPAL



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul - Palani Highway, Dindigul - 624 002.

Department of Computer Science and Engineering

Value added course on *"Solving Problems using Python programming"*

Course Outline

The course aims to impart practical knowledge on "PYTHON PROGRAMMING" to the budding, technical workforce of SSMIET. This program also focuses on exploring and solving various numerical problems. The course helps to gain knowledge in studying various data types like tuples, lists and dictionaries. The targeted audience shall be aware of developing simple applications using modules and packages.

Why Python Programming?

- Python is one of top 5 programming languages today.
- Python is continued to be a favorite option for data scientists who use it for building and using Machine learning applications, scientific computations, and IoT applications.
- Python though is a trendy language, was not in 2013 regulation. Hence attending this course shall be an added benefit for the students of III and IV year.
- Python is considered to be a beginners language since it is concise and easy to read, and can be used for everything from web development to software development and scientific applications.
- Python is free to use, even for the commercial products, because of its OSI-approved open source license.

Application Domain for Python

- Python is widely used in scientific and numeric computing.
- It is used for developing applications related to Internet of Things.
- It is often used as a support language for software development.
- It is also used to build ERP and e-commerce systems.
- It is also used for Data science, Web visualizations & Scraping.

FDP Training Outlines

- Exposure to Python Programming Language.
- Hands-on training to develop numerical applications.
- Training on simple applications creation.
- The Certificate of Participation.
- Mini project competition will be conducted at the completion of the course and the best performer will be honored with a gift.

Participant Registration Fee: Nil

Course Registration Incharge: Ms. N. Padma Priya (Last date for registration: 27/7/2018)

Course Coordinators: Ms. S. Bharathi, Ms. Niveditha, Ms. N. Padma Priya



Dr. D. SENTHIL KUMAR, M.E., Ph.D., F.I.NUS
Head of Department
Department of Mechanical Engineering
SSM Institute of Engineering and Technology
Palani Road, Dindigul 624 002



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL - 624 002

Department of Computer Science and Engineering

Value Added Course - Solving Problem using Python Programming

Attendance Sheet

Date : 06.08.2018 to 14.09.2018

Time: 04:30pm to 05:30pm

Venue: CSE Lab 2

Faculty Incharge : Ms.V.Niveditha AP/CSE

M.S.Bharathi AP/CSE

M.N.Padmapriya AP/CSE

SNo	Reg.no.	Student Name	Month : August-2018												Month : Sep-2018													
			6	7	8	9	10	11	12	13	14	15	16	17	20	21	22	23	24	3	4	5	6	7	10	11	12	13
1	922116104002	Abinaya V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
2	922116104003	Abiramini S	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
3	922116104006	Anush Rathani B	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
4	922116104008	Ashwini A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
5	922116104009	Avanthika SK	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
6	922116104010	Ayswariya Vp	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
7	922116104012	Bharathi S	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
8	922116104013	Bhuvaneswari K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
9	922116104014	Catherine Fransina M	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
10	922116104015	Divya Shree D.K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
11	922116104016	Durga B	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
12	922116104017	Gayathri Varunya B	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
13	922116104018	Gopi Krd	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
14	922116104019	Gowthikka K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
15	922116104022	Hari Prasath M	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
16	922116104023	Kailasan K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
17	922116104025	Kovarthini R	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
18	922116104026	Kulandai Vel K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
19	922116104027	Mahalakshmi P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
20	922116104036	Prarthana S	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
21	922116104038	Roseline Jayashree]	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
22	922116104040	Sameen Farima S	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
23	922116104041	Shalini R	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
24	922116104042	Shangavi R	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
25	922116104043	Sharmugapriya B	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P



Department of Mechanical Engineering
6SM Institute of Engineering Technology

Palani Road, Dindigul-624 002

SSM

INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL - 624 002

Department of Computer Science and Engineering

Value Added Course - Solving Problem using Python Programming

Attendance Sheet

Date : 06.08.2018 to 14.09.2018

Time: 04:30pm to 05:30pm

Venue: CSE Lab 2

Faculty Incharge : Ms.V.Nivedhitha AP/CSE

Ms.S.Bharathi AP/CSE

Ms.N.Padmavathy AP/CSE

Month : August-2018

S.No	Reg.no.	Student Name	6	7	8	9	10	13	14	15	16	17	20	21	22	23	24	3	4	5	6	7	10	11	12	13	14
26	922116104044	Shanmugapriya K	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
27	922116104045	Shifana Sheerin A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
28	922116104046	Sivastinmadhumitha G	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
29	922116104047	Soundharya A	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
30	922116104048	Subalakshmi V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
31	922116104049	Subha S	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
32	922116104050	Sugapriya R	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
33	922116104051	Vijayalakshmi R	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
34	922116104052	Vimala Karthika P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
35	922116104054	Vishnu Priya V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
No. Of Present			35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	34	34	34	34	34	34	34	34	34	
No. Of Absent			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Faculty Sign			SSM																								

*V. V. S.
HOD/CSE*

*D. D. Senthil Kumar
Senior Professor*



Dr. D. SENTHIL KUMARAN M.E.TECH
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Main Road, Dindigul-624 002

LIST, TUPLE AND DICTIONARY

A list is a sequence

- A list is a sequence of values of any type enclosed within square brackets [].
- The values in a list are called **elements** or sometimes **items**.
 - Example: $l1 = [10, "Ram", 20.5, 100, "India"]$
- A list within another list is called a **nested list**.
 - Example: $l2 = [10, "Ram", [30, 70]]$
- A list that contains no elements is called an **empty list**.
 - Example: $l3 = []$

Representation of list elements

Consider the list $l1$. It can be represented as follows.

Positive Indexing →	0	1	2	3	4
List Elements →	10	Ram	20.5	100	India
Negative Indexing →	-5	-4	-3	-2	-1

Consider that the list contains “n” elements.

- The positive index starts with 0 and ranges till $n-1$.
- The negative indexing starts with -1(pointing the last element) and ranges till - n (pointing the first element).

List indices work the same way as string indices:

- Any integer expression can be used as an index.
- Trying to read or write an element that does not exist, will generate **IndexError**.
- If an index has a negative value, it counts backward from the end of the list.

Storing and displaying the list elements / Traversing a list

- To store the elements in an empty list,

```
list=[]
n=int(input("Enter the number of elements:"))
for i in range(n):
    a=input("Enter the list element:")
    list.append(a)
```

- To display the list elements,

```
for i in range(n):
    print(list[i])
```



Dr.D.SENTHIL KUMARAN M.E,Ph.D (NUS)
 Senior Professor
 Department of Mechanical Engineering
 GSM Institute of Engineering Technology
 Palani Road, Dindigul-624 002



List operations

Consider the lists, $l1=[10,20,30]$ and $l2=["Ram"]$.

Operation	Operator	Description	Example	Results
Concatenation	+	Joins two or more lists.	$l1+l2$	[10, 20, 30, "Ram"]
Repetition	*	Duplicates the list elements.	$l2 * 3$	['Ram', 'Ram', 'Ram']
Indexing	[]	Selects a single element pointed by the index.	$l1[0]$ $l1[-2]$	10 20
Slicing	[start : end+1]	Selects a part of the list.	$l1[0:2]$ $l1[1:]$ $l1[: 1]$ $l1[:]$ $l1[-2:-1]$	[10,20] [20,30] 10 [10,20,30] 20

Membership	in	Returns True if the specified element is present in the list; Otherwise returns False.	30 in 11	True
	not in	Returns False if the specified element is present in the list; Otherwise returns True.	40 in 11	False
Length	len(list)	Returns the number of elements present in the list	40 not in 11	True
			30 not in 11	False
			len(11)	3

Lists are mutable

- The elements in a list can be modified in a list. Hence lists are mutable.
- Example:

```
l1=[10,20,30]
l1[0]=50
print(l1)
[50,20,30]
```

The element at the 0th index is changed from 10 to 50.

List methods

Python provides methods that operate on lists. They are as follows.

cmp()

- The method **cmp()** compares elements of two lists.
- Syntax: **cmp(list1, list2)**
- Example

```
list1=[123,'abc']
list2=[123,'abc']
print(cmp(list1, list2))
```

- Output

0

len()

- The method **len()** returns the number of elements in the *list*.
- Syntax: **len(list)**
- Example

```
list1=[123,'xyz','zara']
print(len(list1))
```

- Output

3

max()

- The method **max** returns the elements from the *list* with maximum value.
- Syntax: **max(list)**
- Example

```
list1=[123,'xyz','abc']
print(max(list1))
```

- Output

xyz




Dr. D. SENTHIL KUMARAN M.E., Ph.D, INDS
 Senior Professor
 Department of Mechanical Engineering
 SSM Institute of Engineering Technology
 Palani Road, Dindigul-624 002

min()

- The method **min()** returns the elements from the *list* with minimum value.
- Syntax: **min(list)**
- **Example**

```
list1=[123,'xyz','abc']
print(min(list1))
```
- **Output**
123

list()

- The method **list()** takes sequence types and converts them to lists.
- This is used to convert a given tuple into list.
- Syntax: **list(seq)**
- **Example**

```
t=(123,'xyz','abc')
print(list(t))
```
- **Output**
[123, 'xyz', 'abc']

append()

- The method **append()** appends a specified element into the existing list.
- Syntax: **list.append(element)**
- **Example**

```
list=[123,'xyz','abc']
list.append(100)
print(list)
```
- **Output**
[123, 'xyz', 'abc', 100]

count()

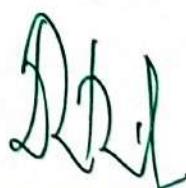
- The method **count()** returns count of how many times *the element* occurs in list.
- Syntax: **list.count(element)**
- **Example**

```
list=[123,'xyz','abc',123]
print(list.count(123))
```
- **Output:**
2

extend()

- The method **extend()** appends the contents of *seq* to list.
- Syntax: **list.extend(newlist)**
- **Example:**

```
list1=[123,'xyz','abc']
list2=[100,'mno']
list1.extend(list2)
print(list1)
```
- **Output:**
[123, 'xyz', 'abc', 100, 'mno']



Dr.D.SENTHIL KUMARAN M.E.,Ph.D (NUS)
Senior Professor
Department of Mechanical Engineering
BSM Institute of Engineering Technology
Palani Road, Dindigul-624 002



index()

- The method **index()** returns the lowest index in list that *obj* appears.
- Syntax: **list.index(element)**
- Example:

```
list=[123,'xyz','abc']
print(list.index('abc'))
```
- Output:
2

insert()

- The method **insert()** inserts a new element into list at the specified *index*.
- Syntax: **list.insert(index, element)**
- Example:

```
list=[123,'xyz','abc']
list.insert(2,100)
print(list)
```
- Output:
[123, 'xyz', 100, 'abc']

pop()

- The method **pop()** removes the last element from the list.
- If the index is passed as parameter, the corresponding element will be popped out.
- Syntax: **list.pop()**
- Example:

```
list=[123,'xyz','abc']
print(list.pop())
print(list.pop(0))
```
- Output:
abc
123

remove()

- This method deletes the element specified from the list.
- Syntax: **list.remove(element)**
- Example:

```
list=[123,'xyz','abc']
list.remove('xyz')
print(list)
```

- Output:

```
[123, 'abc']
```

reverse()

- The method **reverse()** reverses the elements in the list.
- Syntax: **list.reverse()**
- Example:

```
list=[123,'abc', 'xyz']
list.reverse()
print(list)
```

- Output:

```
['xyz', 'abc', 123]
```




Dr. D. SENTHIL KUMARAN M.E., Ph.D (NUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palai Road, Dindigul-624 002

sort()

- The method **sort()** sorts elements of list.
- Syntax: **list.sort()**
- **Example:**

```
list=[123,'xyz','abc']
list.sort()
print(list)
```
- **Output:**

```
[123, 'abc', 'xyz']
```

split()

- The function breaks a string into a list.
- Syntax: **list.split()**
- **Example:**

```
list="python is a language"
list.split()
print(list)
```
- **Output:**

```
['python', 'is', 'a', 'language']
```

join()

- **join()** is the inverse of **split()**.
- It takes a list of strings and concatenates the elements.
- **Example:**

```
list=['python', 'is', 'a', 'language']
a=' '
print(a.join(list))
```
- **Output:**

```
'python is a language'
```

Objects and values

Consider the assignment statements:

```
a = 'xyz'
b = 'xyz'
```

Here a and b both refer to a string, but it is unknown whether they refer to the same string. There are two possible states.

Case 1	Case 2
$a \rightarrow xyz$	$a \rightarrow xyz$
$b \rightarrow xyz$	$b \rightarrow xyz$



Dr.D.SENTHIL KUMARAN M.E.,Ph.D INUS

Senior Professor

Department of Mechanical Engineering

SSM Institute of Engineering Technology

Palani Road, Dindigul-624 002

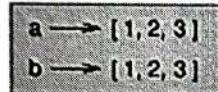
- Case 1: a and b refer to two different objects.
- Case 2: a and b refer to the same object.
- To check whether two variables refer to the same data, **identity operator(is)** can be used.
- **Example:**

```
a = 'xyz'
b = 'xyz'
print(a is b)
```



- **Output:**
True
- Python only created one string object, and both a and b refer to it.
- But when two lists are created, two objects will be obtained.
- **Example:**

```
a = [1, 2, 3]
b = [1, 2, 3]
a is b
```
- **Output:**
False

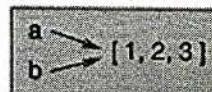


- The two lists are **equivalent**, because they have the same elements,
- But the lists are not **identical**, because they are not the same object.
- If two objects are identical, they are also equivalent.

Aliasing

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

```
a = [1, 2, 3]
b = a
print(b is a)
```
- **Output:**
True



- The association of a variable with an object is called a **reference**.
- In this example, there are two references to the same object.
- An object with more than one reference has more than one name, so we say that the object is **aliased**.
- If the aliased object is mutable, changes made with one alias affect the other.
- **Example:**

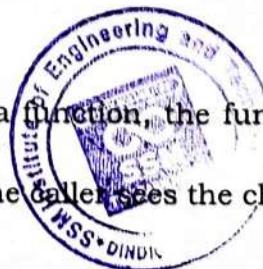
```
a = [1, 2, 3]
b = a
b[0] = 10
print(a)
```

- **Output:**
[10, 2, 3]

List arguments

- When a list is passed as a parameter to a function, the function gets a reference to the list.
- If the function modifies a list parameter, the caller sees the change.
- **Example:**

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



Dr. D. SENTHIL KUMARAN M.E., Ph.D
Senior Professor
Department of Mechanical Engineering
SSN Institute of Engineering & Technology
Perundurai, Madhugiri - 602 102

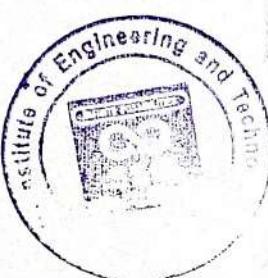
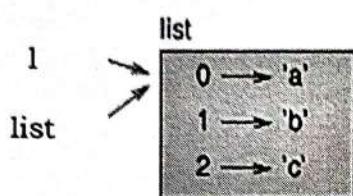
```
l = ['a', 'b', 'c']
```

```
sample(l)
```

```
print(l)
```

- **Output:**

```
['b', 'c']
```



A handwritten signature in black ink, appearing to read "D. Senthil Kumaran".

Dr.D.SENTHIL KUMARAN M.E.Ph.D 2002

Senior Professor

Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

TUPLES

Tuple

- A tuple is a sequence of values of any type enclosed within parenthesis (), separated by comma.
- Providing parenthesis is optional.
- **Example:**

```
tuple1 = ('a', 'b', 'c', 'd', 'e')
tuple2 = (10, 'abc', 'xyz', 20.5)
```

Creating a tuple

Single elemental tuple:

- To create a tuple with a single element, then it is mandatory to include a comma at the end.
- **Example:**

```
t1 = 'a',
print(type(t1))
```
- **Output:**

```
<type 'tuple'>
```

Using built-in function:

- Another way to create a tuple is the built-in function tuple().
- tuple() creates an empty tuple, with no argument.
- **Example:**

```
t1 = tuple()
print(t1)
t2 = tuple("python")
print(t2)
t3 = tuple(10, "Ram", 20.5, 100, "India")
print(t3)
```

- **Output:**

```
()
('p', 'y', 't', 'h', 'o', 'n')
(10, 'Ram', 20.5, 100, 'India')
```

Representation of tuple elements

Consider the tuple t3. It can be represented as follows.

Positive Indexing →	0	1	2	3	4
List Elements →	10	Ram	20.5	100	India
Negative Indexing →	-5	-4	-3	-2	-1

Storing and displaying the tuple elements / Traversing a tuple

- To store the elements in an empty list,

```
t=tuple()
n=int(input("Enter the number of elements:"))
for i in range(n):
    a=input("Enter the tuple element:")
    t.append(a)
```

- To display the tuple elements,
for i range(n):
 print(t[i])



Dr. B. SENTHIL KUMARAN M.E., Ph.D, INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul - 624 092

Tuple operations

Consider the tuples, $t1=(10,20,30)$ and $t2=("Ram")$.

Operation	Operator	Description	Example	Results
Concatenation	+	Joins two or more tuples.	$t1+t2$	[10, 20, 30, "Ram"]
Repetition	*	Duplicates the tuple elements.	$t2 * 3$	['Ram', 'Ram', 'Ram']
Indexing	[]	Selects a single element pointed by the index.	$t1[0]$ $t1[-2]$	10 20
Slicing	[start : end+1]	Selects a part of the tuple.	$t1[0:2]$	[10,20]
			$t1[1:]$	[20,30]
			$t1[: 1]$	10
			$t1[:]$	[10,20,30]
			$t1[-2:-1]$	20
Membership	in	Returns True if the specified element is present in the tuple; Otherwise returns False.	30 in t1	True
			40 in t1	False
	not in	Returns False if the specified element is present in the tuple; Otherwise returns True.	40 not in t1	True
			30 not in t1	False
Length	len(tuple)	Returns the number of elements present in the tuple.	len(t1)	3

Tuples are immutable

- The elements in a tuple cannot be modified once it is assigned. Hence tuples are immutable.
- Example:**

$t1=(10,20,30)$

$t1[0]=50$

TypeError: object doesn't support item assignment

- The only solution is that the modifications can be assigned to a new tuple.

- Example:**

$t1=(10,20,30)$

$t2=(50,) + t1[1:]$

print(t2)

- Output:**

(50, 20, 30)



Dr. D. SENTHIL KUMARAN, M.E., Ph.D (M.Tech.)

Senior Professor

Department of Mechanical Engineering

BSM Institute of Engineering Technology

Palani Road, Dindigul-624 002

Tuple assignment

- The normal method of swapping two values a and b is:

$temp = a$

$a = b$

$b = temp$

- Tuple assignment** reduces the usage of third variable.

$a, b = b, a$

- The left side is a tuple of variables. The right side is a tuple of expressions.

- Each value is assigned to its respective variable.

- The number of variables on the left and the number of values on the right have to be the same.
- Example:
 $a, b = 1, 2, 3$
ValueError: too many values to unpack

Tuples as return values

- A function can only return one value.
- If the value is a tuple, then multiple values can be returned.
- **Example 1:**

```
defdivmod(a,b):
    return(a//b,a%b)
t = divmod(7, 3)
print(t)
```

Output:

(2, 1)

Built-in Tuple Functions

cmp()

- The method **cmp()** compares elements of two tuples.
- Syntax: **cmp(tuple1, tuple2)**
- **Example:**

```
tuple1=(123,'abc')
tuple2=(123,'abc')
print(cmp(tuple1, tuple2))
```

- **Output:**

0

len()

- The method **len()** returns the number of elements in the tuple.
- Syntax: **len(tuple)**
- **Example:**

```
tuple1=(123,'xyz','abc')
print(len(tuple1))
```

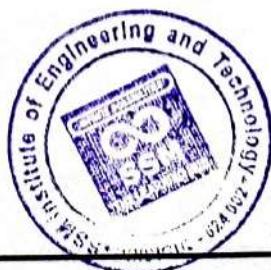
- **Output:**

3

max()

- The method **max()** returns the elements from the tuple with maximum value.
- Syntax: **max(tuple)**
- **Example:**

```
tuple1 =(123,'xyz','abc')
print(max(tuple1))
```



- **Output:**

xyz

min()

- The method **min()** returns the elements from the tuple with minimum value.
- Syntax: **min(tuple)**
- **Example:**

```
tuple1=(123,'xyz','abc')  
print(min(tuple1))
```

- **Output:**

123

tuple()

- The method **tuple()** converts a list of items into tuples.
- Syntax: **tuple(elements)**
- **Example:**

```
a=[123,'xyz','abc']  
t=tuple(a)  
print(t)
```

- **Output:**

(123, 'xyz', 'abc')



Dr.D.SENTHIL KUMARAN M.E.,Ph.D,INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

A handwritten signature in black ink, appearing to read "D. Senthil Kumaran".

DICTIONARIES

Dictionary

- A dictionary as a mapping between a set of **keys** and a set of values separated by comma and enclosed within **curly braces** {}.
- Each key maps to a value.
- The association of a key and a value is called a **key-value pair**.
- **Example:**

```
d={'Name':'Zara','Age':7}  
print(d)
```

- **Output:**

```
{'Age': 7, 'Name': 'Zara'}
```

- The order of the key-value pairs is not the same.
- The function dict() is used to create a new dictionary with no items.

- **Example:**

```
d = dict()  
print(d)
```

- **Output:**

```
{}
```

Dictionary as a set of counters

- To count how many times each letter appears in a string.
- **Example:**

```
def histogram(s):  
    d = dict()  
    for c in s:  
        if c not in d:  
            d[c] = 1  
        else:  
            d[c] = d[c] + 1  
    return d
```

```
h = histogram('october')  
print(h)
```

- **Output:**

```
{'o': 2, 'c': 1, 't': 1, 'b': 1, 'e': 1, 'r': 1}
```

Looping and dictionaries

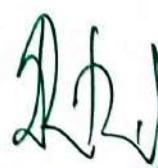
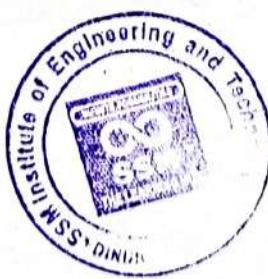
- A dictionary in a for statement, traverses the keys of the dictionary.
- **Example:**

```
def print_hist(h):  
    for c in h:  
        print(c, h[c])  
h = histogram('parrot')  
print_hist(h)
```

- **Output:**

```
a 1  
p 1  
r 2  
t 1  
o 1
```

- The keys do not follow the same order.



Reverse lookup

- Given a dictionary d and a key k, it is easy to find the corresponding value $v = d[k]$. This operation is called a **lookup**. The reverse case is said to be reverse lookup.
- If the value v is given, to find k is have certain problems.
 - There might be more than one key that maps to the value v.
 - There is no simple syntax to do a reverse lookup.

Function that takes a value and returns the first key that maps to that value:

```
defreverse_lookup(d, v):  
    for k in d:  
        if d[k] == v:  
            return k  
  
    raise ValueError
```

Built-in Dictionary Functions & Methods

cmp()

- The method **cmp()** compares two dictionaries based on key and values.
- Syntax: **cmp(dict1, dict2)**
- Example:**

```
dict1 ={'Name':'Zara','Age':7}  
dict2 ={'Name':'Zara','Age':7}  
print(cmp(dict1, dict2))
```

- Output:**

0

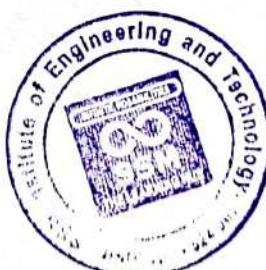
str()

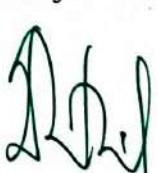
- The method **str()** produces a printable string representation of a dictionary.
- Syntax: **str(dict)**
- Example:**

```
dict={'Name':'Zara','Age':7}  
print(str(dict))
```

- Output:**

{'Age': 7, 'Name': 'Zara'}




Dr. D. SENTHIL KUMARAN M.E., Ph.D., INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

type()

- The method **type()** returns the type of the specified variable. If the variable is dictionary then it would return a dictionary type.
- Syntax: **type(dict)**
- Example:**

```
dict={'Name':'Zara','Age':7}
```

```
print( type (dict))
```

- **Output:**

Variable Type : <type 'dict'>

clear()

- The method **clear()** removes all items from the dictionary.

- Syntax: **dict.clear()**

- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(len(dict))  
dict.clear()  
print(len(dict))
```

- **Output:**

2

0

copy()

- The method **copy()** returns a copy of the dictionary.

- Syntax: **dict.copy()**

- **Example:**

```
dict1 ={'Name':'Zara','Age':7}  
dict2 =dict1.copy()  
print(dict2)
```

- **Output:**

{'Age': 7, 'Name': 'Zara'}

fromkeys()

- The method **fromkeys()** creates a new dictionary with keys and *values*.

- Syntax: **dict.fromkeys(key, value)**

- **Example:**

```
key=('name','age')  
dict=dict.fromkeys(key)  
print(str(dict))  
dict=dict.fromkeys(key,10)  
print(str(dict))
```

- **Output:**

{'age': None, 'name': None}
'age': 10, 'name': 10}




Dr. D. SENTHIL KUMARAN M.E., Ph.D., M.N.S.
Senior Professor
Department of Mechanical Engineering
SSN Institute of Engineering Technology
Main Road, Jindigai - 624 002

get()

- The method **get()** returns a value for the given key. If key is not available then returns default value None.
- Syntax: **dict.get(key, default = None)**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.get('Age'))
```

Output:

7

has_key()

- The method **has_key()** returns True if a given key is available in the dictionary, otherwise it returns a False.
- Syntax: **dict.has_key(key)**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.has_key('Age'))
```

- **Output:**

True

items()

- The method **items()** returns a list of dict's (key, value) tuple pairs
- Syntax: **dict.items()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.items())
```

- **Output:**

[(‘Age’, 7), (‘Name’, ‘Zara’)]

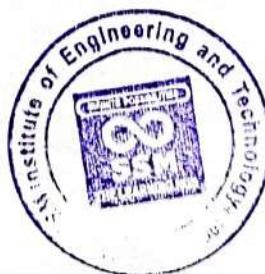
keys()

- The method **keys()** returns a list of all the available keys in the dictionary.
- Syntax: **dict.keys()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.keys())
```

- **Output:**

['Age', 'Name']




Dr. D. SENTHIL KUMARAN M.E., Ph.D (INDIA)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

setdefault()

- The method **setdefault()** is similar to **get()**, but will set **dict[key]=default** if key is not already in dict.
- Syntax: **dict.setdefault(key, default value)**
- **Example:**

```
dict={'Name':'Zara'}
print(dict.setdefault('Age',None))
```

- **Output:**

None

update()

- The method **update()** adds dictionary **dict2's** key-values pairs in to **dict**. This function does not return anything.
- Syntax: **dict.update(dict2)**
- **Example:**

```
dict={'Name':'Zara','Age':7}
dict2 ={'mark':100}
dict.update(dict2)
print(dict)
```

- **Output:**

{'Age': 7, 'Name': 'Zara', 'mark': 100}

values()

- The method **values()** returns a list of all the values available in a given dictionary.
- Syntax: **dict.values()**
- **Example:**

```
dict={'Name':'Zara','Age':7}
print(dict.values())
```

- **Output:**

[7, 'Zara']




Dr. D. SENTHIL KUMARAN M.E., Ph.D. (INUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul - 624 001

LISTS IN PYTHON

A list is a sequence

- A **list** is a sequence of values of any type enclosed within square brackets [].
- The values in a list are called **elements** or sometimes **items**.
 - Example: `l1 = [10, "Ram", 20.5, 100, "India"]`
- A list within another list is called a **nested list**.
 - Example: `l2 = [10, "Ram", [30, 70]]`
- A list that contains no elements is called an **empty list**.
 - Example: `l3 = []`

Representation of list elements

Consider the list `l1`. It can be represented as follows.

Positive Indexing →	0	1	2	3	4
List Elements →	10	Ram	20.5	100	India
Negative Indexing →	-5	-4	-3	-2	-1

Consider that the list contains “n” elements.

- The positive index starts with 0 and ranges till $n-1$.
- The negative indexing starts with -1 (pointing the last element) and ranges till $-n$ (pointing the first element).

List indices work the same way as string indices:

- Any integer expression can be used as an index.
- Trying to read or write an element that does not exist, will generate **IndexError**.
- If an index has a negative value, it counts backward from the end of the list.

Storing and displaying the list elements / Traversing a list

- To store the elements in an empty list,

```
list=[]
n=int(input("Enter the number of elements:"))
for i in range(n):
    a=input("Enter the list element:")
    list.append(a)
```

- To display the list elements,

```
for i in range(n):
    print(list[i])
```

List operations

Consider the lists, `l1=[10,20,30]` and `l2=["Ram"]`.

Operation	Operator	Description	Example	Results
Concatenation	+	Joins two or more lists.	<code>l1+l2</code>	[10, 20, 30, "Ram"]
Repetition	*	Duplicates the list elements.	<code>l2 * 3</code>	['Ram','Ram','Ram']
Indexing	[]	Selects a single element pointed by the index.	<code>l1[0]</code> <code>l1[-2]</code>	10 20
Slicing	<code>[start :end+1]</code>	Selects a part of the list.	<code>l1[0:2]</code> <code>l1[1:]</code> <code>l1[:1]</code> <code>l1[:]</code>	[10,20] [20,30] 10 [10,20,30]

Dr.D.SENTHILKUMARAN M.E. Ph.D IAS

Senior Professor



LISTS IN PYTHON

Membership	in	Returns True if the specified element is present in the list; Otherwise returns False.	30 in l1	True
	not in	Returns False if the specified element is present in the list; Otherwise returns True.	40 not in l1	True
Length	len(list)	Returns the number of elements present in the list	30 not in l1	False
			len(l1)	3

Lists are mutable

- The elements in a list can be modified in a list. Hence lists are mutable.
- Example:

```
l1=[10,20,30]
l1[0]=50
print(l1)
[50,20,30]
```

The element at the 0th index is changed from 10 to 50.

List methods

Python provides methods that operate on lists. They are as follows.

cmp()

- The method **cmp()** compares elements of two lists.
- Syntax: **cmp(list1, list2)**
- Example

```
list1=[123,'abc']
list2=[123,'abc']
print(cmp(list1, list2))
```

- Output**

0

len()

- The method **len()** returns the number of elements in the *list*.
- Syntax: **len(list)**
- Example

```
list1=[123,'xyz','zara']
print(len(list1))
```

- Output**

3

max()

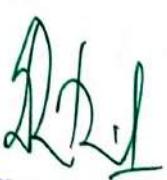
- The method **max** returns the elements from the *list* with maximum value.
- Syntax: **max(list)**
- Example

```
list1=[123,'xyz','abc']
print(max(list1))
```

- Output**

xyz




Dr. D. SENTHIL KUMARAN M.E., Ph.D (IITB)
 Senior Professor
 Department of Mechanical Engineering
 BSM Institute of Engineering Technology
 Palani Road, Dindigul - 624 002

LISTS IN PYTHON

min()

- The method **min()** returns the elements from the *list* with minimum value.
- Syntax: **min(list)**
- Example**

```
list1=[123,'xyz','abc']
print(min(list1))
```
- Output**
123

list()

- The method **list()** takes sequence types and converts them to lists.
- This is used to convert a given tuple into list.
- Syntax: **list(seq)**
- Example**

```
t=(123,'xyz','abc')
print(list(t))
```
- Output**
[123, 'xyz', 'abc']

append()

- The method **append()** appends a specified element into the existing list.
- Syntax: **list.append(element)**
- Example**

```
list=[123,'xyz','abc']
list.append(100)
print(list)
```
- Output**
[123, 'xyz', 'abc', 100]

count()

- The method **count()** returns count of how many times *the element* occurs in list.
- Syntax: **list.count(element)**
- Example**

```
list=[123,'xyz','abc',123]
print(list.count(123))
```
- Output:**
2

extend()

- The method **extend()** appends the contents of *seq* to list.
- Syntax: **list.extend(newlist)**
- Example:**

```
list1=[123,'xyz','abc']
list2=[100,'mno']
list1.extend(list2)
print(list1)
```
- Output:**
[123, 'xyz', 'abc', 100, 'mno']

Dr.D.SENTHIL KUMARAN M.E.,P.G INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002



LISTS IN PYTHON

index()

- The method **index()** returns the lowest index in list that *obj* appears.
- Syntax: **list.index(element)**
- Example:

```
list=[123,'xyz','abc']
print(list.index('abc'))
```
- Output:
2

insert()

- The method **insert()** inserts a new element into list at the specified *index*.
- Syntax: **list.insert(index, element)**
- Example:

```
list=[123,'xyz','abc']
list.insert(2,100)
print(list)
```
- Output:
[123, 'xyz', 100, 'abc']

pop()

- The method **pop()** removes the last element from the list.
- If the index is passed as parameter, the corresponding element will be popped out.
- Syntax: **list.pop()**
- Example:

```
list=[123,'xyz','abc']
print(list.pop())
print(list.pop(0))
```
- Output:
abc
123

remove()

- This method deletes the element specified from the list.
- Syntax: **list.remove(element)**
- Example:

```
list=[123,'xyz','abc']
list.remove('xyz')
print(list)
```
- Output:
[123, 'abc']

reverse()

- The method **reverse()** reverses the elements in the list.
- Syntax: **list.reverse()**
- Example:

```
list=[123,'abc', 'xyz']
list.reverse()
print(list)
```
- Output:
['xyz', 'abc', 123]



Dr. D. SENTHIL KUMARAN M.E., Ph.D, INUS

Senior Professor

Department of Mechanical Engineering

SSM Institute of Engineering Technologies

Palau Road, Thudigai-624 002



LISTS IN PYTHON

sort()

- The method **sort()** sorts elements of list.
- Syntax: **list.sort()**
- **Example:**

```
list=[123,'xyz','abc']
list.sort()
print(list)
```

- **Output:**

```
[123, 'abc', 'xyz']
```

split()

- The function breaks a string into a list.
- Syntax: **list.split()**
- **Example:**

```
list="python is a language"
list.split()
print(list)
```

- **Output:**

```
['python','is','a','language']
```

join()

- **join()** is the inverse of **split()**.
- It takes a list of strings and concatenates the elements.
- **Example:**

```
list=['python','is','a','language']
a=' '
print(a.join(list))
```

- **Output:**

```
'python is a language'
```

EXERCISE PROGRAMS

Sum of the elements in a list

```
list1=[10,20,30,40,50]
sum=0
for n in list1:
    sum=sum+n
print("Sum of the elements in the list is ",sum)
```

Output

```
Sum of elements in the list is 150
```

Sum of the elements in a list - Getting input from the user

```
list1=[]
print("Enter the number of elements in the list : ")
n=int(input())
print("Enter the list elements : ")
for i in range(n):
    elt=int(input())
    list1.append(elt)
sum=0
for num in list1:
    sum=sum+num
print("Sum of the elements in the list is ",sum)
```

Output

```
Enter the number of elements in the list : 5
Enter the list elements :
```

```
10
20
30
40
50
```

Dr.D.SENTHIL KUMARAN M.E.P.D INUS

Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

LISTS IN PYTHON

Finding unique elements in a list

```
list1 = [10, 20, 10, 30, 40, 40]
unique=[]
for elt in list1:
    if(elt not in unique):
        unique.append(elt)

print("The unique elements are :")
for elt in unique:
    print(elt)
```

Output:

The unique elements are :

10
20
30
40

Finding unique elements in a list – Get the input from the user

```
list1=[]
print("Enter the number of elements in the list : ")
n=int(input())
print("Enter the list elements : ")
for i in range(n):
    elt=input()
    list1.append(elt)
unique=[]
for i in list1:
    if(i not in unique):
        unique.append(i)
print("The unique elements are :")
for i in unique:
    print(i)
```

Output:

Enter the number of elements in the list : 5

Enter the list elements :

9221
CSE
9221
50.5
CSE

The unique elements are :

9221
CSE
50.5

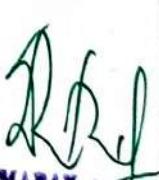
Finding the maximum and minimum element in the list

```
list1=[]
print("Enter the number of elements in the list : ")
n=int(input())
print("Enter the list elements : ")
for i in range(n):
    a=int(input())
    list1.append(a)
max,min=list1[0],list1[0]

for elt in list1:
    if(min>elt):
        min=elt
    if(max<elt):
        max=elt

print("The maximum element is {} and minimum element is {}".format(max,min))
minpos = list1.index(min)
maxpos = list1.index(max)
print("The maximum element is found at {} position and minimum element is found at {} position".format(maxpos,minpos))
```




Dr. D. SENTHIL KUMARAN M.E., Ph.D., IITB
Senior Professor
Department of Mechanical Engineering
SSN Institute of Engineering Technology
Palani Road, Dindigul-624 002

Output

Enter the number of elements in the list :5

Enter the list elements :

45

22

99

33

11

The maximum element is 99 and minimum element is 11.

The maximum element is found at 2 position and minimum element is found at 4 position

Number of occurrences of an element in list

```
list1=[]
print("Enter the number of elements in the list : ")
n=int(input())
print("Enter the list elements : ")
for i in range(n):
    a=int(input())
    list1.append(a)
count=0
m=int(input("Enter the element to be found : "))
for elt in list1:
    if(elt == m):
        count = count + 1
print("{} has occurred {} times".format(m,count))
```

Output

Enter the number of elements in the list : 7

Enter the list elements :

5

2

8

5

1

6

5

5

Enter the element to be found : 5

5 has occurred 3 times

Linear Search

```
list=[]
n=int(input("Enter number of elements: "))
print("Enter the list elements:")
for i in range(0,n):
    a=int(input())
    list.append(a)
key=int(input("Enter the element to be searched: "))
for elt in list:
    if(elt == key):
        print("The element is present at {}th position".format(i+1))
        break
else:
    print("The element is not present in the list")
```

Output:

Enter number of elements: 5

Enter the list elements:

6

1

8

4

3

Enter the element to be searched: 4

The element is present at 5th position




Dr. D. SENTHIL KUMARAN M.E., PH.D
Senior Professor
Department of Mechanical Engineering
BSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

LISTS IN PYTHON

Binary Search

```
list=[]
n=int(input("Enter number of elements: "))
print("Enter the list elements:")
for i in range(0,n):
    a=int(input())
    list.append(a)
key=int(input("Enter the element to be searched: "))
first=0
last=n-1
b=False
while(first<=last):
    mid=(first + last)//2
    if(list[mid]==key):
        b=True
        break
    elif(key < list[mid]):
        last = mid - 1
    else:
        first = mid + 1
if(b==True):
    print("The element is found")
else:
    print("The element is not found")
```

Output:

Enter number of elements: 5

Enter the list elements:

6

1

8

4

3

Enter the element to be searched: 4

The element is found


Dr. D. SENTHIL KUMARAN M.E., Ph.D, IITB
Senior Professor
Department of Mechanical Engineering
ESM Institute of Engineering Technology
Palani Road, Tanjore 524 002

Objects and values

Consider the assignment statements:

```
a = 'xyz'  
b = 'xyz'
```

Here a and b both refer to a string, but it is unknown whether they refer to the same string. There are two possible states.

Case 1	Case 2
a → xyz	a → xyz
b → xyz	b → xyz

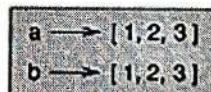
- Case 1: a and b refer to two different objects.
- Case 2: a and b refer to the same object.
- To check whether two variables refer to the same data, **identity operator (is)** can be used.
- **Example:**

```
a = 'xyz'  
b = 'xyz'  
print(a is b)
```
- **Output:**
True
- Python only created one string object, and both a and b refer to it.

Reference:

- The association of a variable with an object is called a reference.
- When two lists are created, two objects will be obtained.
- To check whether two variables refer to the same data, **identity operator (is)** can be used.
- **Example:**

```
a = [1, 2, 3]  
b = [1, 2, 3]  
a is b
```
- **Output:**
False



Aliasing

- Two variables refers to same object is said to be aliasing



- Aliasing can be achieved by assigning the one object to another.
- Thus, the assignment of one object to another creates a binding and both variables refer to the same object.
- **Example:**

```
a = [1, 2, 3]  
b = a  
print(b is a)
```
- **Output:**
True
- In this example, there are two references to the same object.
- An object with more than one reference has more than one name, so we say that the object is aliased.

Dr.D.SENTHIL KUMARAN M.E.,Ph.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002



- If the aliased object is mutable, changes made with one alias affect the other.
- **Example:**

```
a = [1, 2, 3]
b = a
b[0] = 10
print(a)
```
- **Output:**

```
[10, 2, 3]
```

Cloning

- Two variables obtain references to copies of equivalent objects is said to be cloning
- When a list is copied to another list, it is said to be cloned.
- Now, the two lists are **equivalent**, because they have the same elements, but the lists are not **identical**, because they are not the same object.
- Cloning is attained by assigning list with slicing operator or by using built-in methods

a → [1, 2, 3]
b → [1, 2, 3]

- **Example:**

```
a=[1,2,3]
b=a[:]
print(a)
print(b)
b[2]=5
print(b)
```
- **Output:**

```
[1,2,3]
[1,2,3]
[1,2,3]
[1,2,5]
```

- The changes made on a cloned list, will not reflect in the original list.

Cloning can be achieved in two ways using built in methods.

- list()
- copy()
- **list()** method is used to clone a list by the following syntax.
 - Syntax: newlist=list(oldlist)
 - **Example:**

```
a=[1,2,3]
b=list(a)
a[0]=100
b[0]=200
print(a)
print(b)
```

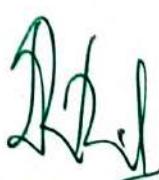
- **Output:**

```
[100,2,3]
[200,2,3]
```

- **copy()** method can be used as follows.
 - Syntax: newlist=copy(oldlist)
 - **Example:**

```
from copy import *
a=[1,2,3]
b=copy(a)
a[0]=100
b[0]=200
print(a)
print(b)
```




Dr. D. SENTHIL KUMARAN M.E., Ph.D., IITB
 Senior Professor
 Department of Mechanical Engineering
 SSM Institute of Engineering Technology
 Palani Road, Dindigul-624 002

- **Output:**
[100,2,3]
[200,2,3]

List Comprehension

- List comprehensions provide a short way to create lists from existing lists.
- It creates a new list where each element is the result of some operations applied to each member of existing list that satisfy some conditions.
- Example: **list of squares**

```
s = []
for x in range(10):
    s.append(x**2)
print(s)
```

Output:

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

- The above code creates or overwrites a variable named x that still exists after the loop completes.
- The list comprehension creates the new list without any side effects.
- This is achieved by
 - Syntax: **variable=[var for var in sequence]**
 - Example: **s = [x**2 for x in range(10)]**
- A list comprehension consists of brackets containing an expression followed by a **for** clause, then zero or more **for** or **if** clauses.
- The result will be a new list resulting from evaluating the expression in the context of the **for** and **if** clauses which follow it.

Sample program - 1:

```
s=[x**2 for x in range(10)]
print(s)
```

Output:

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

Sample program - 2:

```
s=[x**2 for x in range(10) if x%2 ==0]
print(s)
```

Output:

[0, 4, 16, 36, 64]

MAP, FILTER and REDUCE functions

map()

- Map applies a function to all the items in an input list.
- Syntax: **map(function,sequence)**
- Example - 1:

```
items=[1, 2, 3, 4, 5]
s=list(map(lambda x: x**2, items))
print(s)
```

- Output:
[1, 4, 9, 16, 25]

- Example - 2:
items=[3,5,-1,8,0]


```
s=list(map(lambda x:x>0, items))
print(s)
```

- Output:
[True, True, False, True, False]



Dr.D.SENTHIL KUMARAN M.E., Ph.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

filter()

- filter() creates a list of elements for which a function returns True.
- Syntax: **filter(function, sequence)**

- **Example - 1:**

```
items=[1, 2, 3, 4, 5]
s=list(filter(lambda x: x%2==0, items))
print(s)
```

- **Output:**

[2, 4]

- **Example - 2:**

```
items=[3,5,-1,8,0]
s=list(filter(lambda x:x>0, items))
print(s)
```

- **Output:**

[3,5,8]

reduce()

- This function applies a rolling computation to sequential pairs of values in a list.

- Syntax: **reduce(function, sequence)**

- **Example - 1:**

```
>>>fromfunctools import reduce
>>>reduce(lambda x, y: x+y, range(1,10))
```

- **Output:**

45

- **Example - 2:**

```
importfunctools
list = [ 1, 3, 5, 6, 2, ]
print ("Sum : ")
print (functools.reduce(lambda a,b : a+b,list))
```

- **Output:**

Sum : 17

LIST'




Dr. D. SENTHIL KUMARAN M.E., Ph.D., INUS
Senior Professor
Department of Mechanical Engineering
BSM Institute of Engineering Technologies
Palani Road, Dindigul-624 002

TUPLES

Tuple

- A tuple is a sequence of values of any type enclosed within parenthesis (), separated by comma.
- Providing parenthesis is optional.
- **Example:**

```
tuple1 = ('a', 'b', 'c', 'd', 'e')
tuple2 = (10, 'abc', 'xyz', 20.5)
```

Creating a tuple

Single elemental tuple:

- To create a tuple with a single element, then it is mandatory to include a comma at the end.
- **Example:**
t1 = 'a',
print(type(t1))
- **Output:**
<type 'tuple'>

Using built-in function:

- Another way to create a tuple is the built-in function tuple().
- tuple() creates an empty tuple, with no argument.
- **Example:**

```
t1 = tuple()
print(t1)
t2 = tuple("python")
print(t2)
t3 = tuple(10, "Ram", 20.5, 100, "India")
print(t3)
```

- **Output:**
(
'p', 'y', 't', 'h', 'o', 'n'
(10, 'Ram', 20.5, 100, 'India')

Representation of tuple elements

Consider the tuple t3. It can be represented as follows.

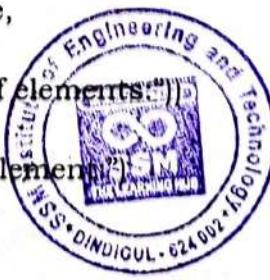
Positive Indexing →	0	1	2	3	4
List Elements →	10	Ram	20.5	100	India
Negative Indexing →	-5	-4	-3	-2	-1

Storing and displaying the tuple elements / Traversing a tuple

- To store the elements in an empty tuple,

```
t=tuple()
n=int(input("Enter the number of elements"))
for i in range(n):
    a=input('Enter the tuple element')
    t.append(a)
```

- To display the tuple elements,
for i in range(n):
 print(t[i])



Dr.D.SENTHIL KUMARAN M.E., Ph.D (INUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

Tuple operations

Consider the tuples, $t1=(10,20,30)$ and $t2=("Ram")$.

Operation	Operator	Description	Example	Results
Concatenation	+	Joins two or more tuples.	$t1+t2$	(10, 20, 30, "Ram")
Repetition	*	Duplicates the tuple elements.	$t2 * 3$	('Ram','Ram','Ram')
Indexing	[]	Selects a single element pointed by the index.	$t1[0]$	10
			$t1[-2]$	20
		Selects a part of the tuple.	$t1[0:2]$	(10,20)
			$t1[1:]$	(20,30)
			$t1[: 1]$	10
			$t1[:]$	(10,20,30)
			$t1[-2:-1]$	20
		Returns True if the specified element is present in the tuple; Otherwise returns False.	30 in $t1$	True
			40 in $t1$	False
Membership	in	Returns False if the specified element is present in the tuple; Otherwise returns True.	40 not in $t1$	True
			30 not in $t1$	False
Length	<code>len(tuple)</code>	Returns the number of elements present in the tuple.	<code>len(t1)</code>	3

Tuples are immutable

- The elements in a tuple cannot be modified once it is assigned. Hence tuples are immutable.
- Example:**

```
t1=(10,20,30)
t1[0]=50
```

TypeError: object doesn't support item assignment

- The only solution is that the modifications can be assigned to a new tuple.
- Example:**

```
t1=(10,20,30)
t2=(50,) + t1[1: ]
print(t2)
```

- Output:**

```
(50, 20, 30)
```

Tuple assignment

- The normal method of swapping two values a and b is.
- ```
temp = a
a = b
b = temp
```
- Tuple assignment** reduces the usage of third variable.
- ```
a, b = b, a
```
- The left side is a tuple of variables. The right side is a tuple of expressions.
 - Each value is assigned to its respective variable.



Dr.D.SENTHIL KUMARAN M.E.,Ph.D, INUS
Senior Professor
Department of Mechanical Engineering
BSM Institute of Engineering Technology
Palani Road, Tiruchirappalli - 624 002

- The number of variables on the left and the number of values on the right have to be the same.
- Example:

a, b = 1, 2, 3

ValueError: too many values to unpack

Built-in Tuple Functions

cmp()

- The method **cmp()** compares elements of two tuples.
- Syntax: **cmp(tuple1, tuple2)**
- Example:

```
tuple1=(123,'abc')
tuple2=(123,'abc')
print(cmp(tuple1, tuple2))
```

- Output:

0

len()

- The method **len()** returns the number of elements in the tuple.
- Syntax: **len(tuple)**
- Example:

```
tuple1=(123,'xyz','abc')
print(len(tuple1))
```

- Output:

3

max()

- The method **max()** returns the elements from the tuple with maximum value.
- Syntax: **max(tuple)**
- Example:

```
tuple1 =(123,'xyz','abc')
print(max(tuple1))
```

- Output:

xyz

min()

- The method **min()** returns the elements from the tuple with minimum value.
- Syntax: **min(tuple)**
- Example:

```
tuple1=(123,'xyz','abc')
print(min(tuple1))
```

- Output:

123



Dr.D.SENTHIL KUMARAN M.E.PH.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology,
Palani Road, Dindigul-624 002

tuple()

- The method **tuple()** converts a list of items into tuples.
- Syntax: **tuple(elements)**
- **Example:**

```
a=[123,'xyz','abc']  
t=tuple(a)  
print(t)
```

- **Output:**

(123, 'xyz', 'abc')

count()

- The method **count()** returns count of how many times *the element* occurs in tuple.
- Syntax: **tuple.count(element)**
- **Example**

```
tuple=(123,'xyz','abc',123)  
print(tuple.count(123))
```

- **Output:**

2

index()

- The method **index()** returns the lowest index in tuple that *obj* appears.
- Syntax: **tuple.index(element)**
- **Example:**

```
tuple=[123,'xyz','abc']  
print(tuple.index('abc'))
```

- **Output:**

2




Dr. D. SENTHIL KUMARAN M.E., Ph.D (I.M.S.)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

DICTIONARIES

Dictionary

- A dictionary as a mapping between a set of **keys** and a set of values separated by comma and enclosed within **curly braces { }**.
- Each key maps to a value.
- The association of a key and a value is called a **key-value pair**.

• Example:

```
d={'Name':'Zara','Age':7}  
print(d)
```

• Output:

```
{'Age': 7, 'Name': 'Zara'}
```

- The order of the key-value pairs is not the same.
- The function dict() is used to create a new dictionary with no items.

• Example:

```
d = dict()  
print(d)
```

• Output:

```
{}
```

Built-in Dictionary Functions & Methods

cmp()

- The method **cmp()** compares two dictionaries based on key and values.
- Syntax: **cmp(dict1, dict2)**
- Example:

```
dict1 ={'Name':'Zara','Age':7}  
dict2 ={'Name':'Zara','Age':7}  
print(cmp(dict1, dict2))
```

• Output:

```
0
```

str()

- The method **str()** produces a printable string representation of a dictionary.
- Syntax: **str(dict)**
- Example:

```
dict={'Name':'Zara','Age':7}  
print(str(dict))
```

• Output:

```
{'Age': 7, 'Name': 'Zara'}
```

type()

- The method **type()** returns the type of the specified variable. If the variable is dictionary then it would return a dictionary type.
- Syntax: **type(dict)**



Dr.D.SENTHIL KUMARAN M.E.Ph.D (NUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

- **Example:**

```
dict={'Name':'Zara','Age':7}  
print( type (dict))
```

- **Output:**

Variable Type : <type 'dict'>

clear()

- The method **clear()** removes all items from the dictionary.
- Syntax: **dict.clear()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(len(dict))  
dict.clear()  
print(len(dict))
```

- **Output:**

2
0

copy()

- The method **copy()** returns a copy of the dictionary.
- Syntax: **dict.copy()**
- **Example:**

```
dict1 ={'Name':'Zara','Age':7}  
dict2 =dict1.copy()  
print(dict2)
```

- **Output:**

{'Age': 7, 'Name': 'Zara'}

fromkeys()

- The method **fromkeys()** creates a new dictionary with keys and *values* .
- Syntax: **dict.fromkeys(key, value)**
- **Example:**

```
key=('name','age')  
dict=dict.fromkeys(key)  
print(str(dict))  
dict=dict.fromkeys(key,10)  
print(str(dict))
```

- **Output:**

{'age': None, 'name': None}
{'age': 10, 'name': 10}



Dr.D.SENTHIL KUMARAN M.E.Ph.D (NUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul 624 002

get()

- The method **get()** returns a value for the given key. If key is not available then returns default value None.
- Syntax: **dict.get(key, default = None)**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.get('Age'))
```

Output:

7

has_key()

- The method **has_key()** returns True if a given key is available in the dictionary, otherwise it returns a False.
- Syntax: **dict.has_key(key)**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.has_key('Age'))
```

- **Output:**

True

items()

- The method **items()** returns a list of dict's (key, value) tuple pairs
- Syntax: **dict.items()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.items())
```

- **Output:**

[(‘Age’, 7), (‘Name’, ‘Zara’)]

keys()

- The method **keys()** returns a list of all the available keys in the dictionary.
- Syntax: **dict.keys()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.keys())
```

- **Output:**

[‘Age’, ‘Name’]



Dr.D.SENTHIL KUMARAN M.E.P.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

setdefault()

- The method **setdefault()** is similar to `get()`, but will set `dict[key]=default` if key is not already in dict.
- Syntax: **dict.setdefault(key, default value)**
- **Example:**

```
dict={'Name':'Zara'}  
print(dict.setdefault('Age',None))
```

- **Output:**

None

update()

- The method **update()** adds dictionary `dict2`'s key-values pairs in to `dict`. This function does not return anything.
- Syntax: **dict.update(dict2)**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
dict2 ={'mark':100}  
dict.update(dict2)  
print(dict)
```

- **Output:**

{Age: 7, 'Name': 'Zara', 'mark': 100}

values()

- The method **values()** returns a list of all the values available in a given dictionary.
- Syntax: **dict.values()**
- **Example:**

```
dict={'Name':'Zara','Age':7}  
print(dict.values())
```

- **Output:**

[7, 'Zara']



Dr.D.SENTHIL KUMARAN M.E.PH.D (NUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

PYTHON ASSIGNMENT

Program:

```
l = [ ]
```

```
a = 0
```

```
for i in range (1, 150):
```

```
    l.append(a)
```

```
    a = a + 1
```

```
n = int(input("Enter the divisor element:"))
```

```
x = 0
```

```
for i in l:
```

```
    if (i % n == 0):
```

```
        x = x + 1
```

```
    if (i == n):
```

```
        x = x - 1
```

print("The total no. of elmts divisible by {} is {}".format(n, x))

Output:

Enter the divisor element: 5

The total no. of elements divisible by 5 is 9.



Dr. D. SENTHIL KUMARAN M.E., PH.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

Submitted by,

Catherine Fransina.M

922116104014

II - CSE

PYTHON ASSIGNMENT

Program to print the value of count, for numbers being divisible from list.

L1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]

```
a = (int)(input ("Enter the no to be divided by:"))
```

```
for (i in L1):
```

```
    if (list[i] % a == 0):
```

```
        print (list[i])
```

```
        count += 1
```

```
    elif (i == a):
```

```
        count -= 1
```

```
print(count)
```

Output

Enter the no to be divided by : 10

20

4

30

40

50



Dr.D.SENTHIL KUMARAN M.E.P.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

submitted By,
V.Vishnu Priya
III - CSE

ASSIGNMENT ON PYTHON

```
L1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17  
      18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,  
      30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,  
      42, 43, 44, 45, 46, 47, 48, 49, 50]  
  
a = int(input("enter the no to be divided"))  
for (i in u)  
    if (list[i] % a == 0):  
        print(list[i])  
        count += 1  
    if (i == a)  
        count -= 1  
print(count)
```



Dr.D.SENTHIL KUMARAN M.E.PH.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

Submitted by.
S.R.Aravanthiga,
CSE - II year

```
L1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
      21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
      37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]
```

```
a = int(input("Enter the number to be divided  
by :"))
```

```
for i in L1
```

```
if (list[i] % a == 0):
```

```
    print(list[i])
```

```
    count += 1
```

```
if (i == a):
```

```
    count -= 1
```

```
print(count):
```




Dr. D. SENTHIL KUMARAN M.E. Ph.D (NUS)
 Senior Professor
 Department of Mechanical Engineering
 SSM Institute of Engineering Technology
 Palani Road, Dindigul-624 002

Assignment.

```
List = []
```

```
a = 0
```

```
for i in range(1, 50):
```

```
    List.append(a)
```

```
    a = a + 1
```

```
n = int(input("Enter divisor element:"))
```

```
x = 0
```

```
for i in list:
```

```
    if (i % n == 0):
```

```
        x = x + 1
```

```
    if (i == n):
```

```
        x = x - 1
```

~~print ("The total no. of elements is: {}".format(x))~~

Output:

Enter divisor element: 5

The total no. of elements is: 9

Dr.D.SENTHIL KUMARAN M.E.,Ph.D (NUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002



Python assignment .

K. Shanmugapriya
119 year CSE.

H = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]

a = (int)(input("Enter the no to be divided by :"))

```
for (i in H)
    if (list[i]/a == 0):
        count += 1
    elif (i==a):
        count -= 1
    print(count)
```



A handwritten signature in black ink, appearing to read "D. Senthil Kumaran".

Dr.D.SENTHIL KUMARAN M.E.P.D.I.M.S.
Senior Professor
Department of Mechanical Engineering
SSN Institute of Engineering Technology
Palani Road, Dindigul-624 002



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY,DINDIGUL-624 002

Department of Computer Science and Engineering

Value Added Course-Solving problem using Python Programming

Evaluation Sheet

S.NO	Reg No	Name	Mark(25)
1	922116104002	Abinaya.V	20
2	922116104003	Abirami.S	24
3	922116104006	Anush Rathna.B	22
4	922116104008	Ashwini.A	21
5	922116104009	Avanthiga.AK	23
6	922116104010	Asywariyaa.VP	20
7	922116104012	Bharathi.S	24
8	922116104013	Bhuvaneshwari.K	21
9	922116104014	Catherine Fransina.M	23
10	922116104015	Divya Shree.DK	24
11	922116104016	Durga.B	22
12	922116104017	Gayathri Varunya.B	25
13	922116104018	Gopi.KRD	23
14	922116104019	Gowshikka.K	24
15	922116104022	Hari Prasath.M	21
16	922116104023	Kailasam.K	20
17	922116104025	Kovarthini.R	24
18	922116104026	Kulandai Vel.K	23
19	922116104027	Mahalakshmi.P	21
20	922116104036	Prarthana.S	21
21	922116104038	Roseline Jayashree.J	20
22	922116104040	Samreen Fathima.S	22
23	922116104041	Shalini.R	24
24	922116104042	Shangavi.R	23
25	922116104043	Shanmugapriya.B	22
26	922116104044	Shanmugapriya.K	23
27	922116104045	Shifana Sheerin.A	22
28	922116104046	Sivasrimadhuumitha.G	24
29	922116104047	Soundharya.A	21
30	922116104048	Subalakshmi.V	22
31	922116104049	Suba.S	24
32	922116104050	Sugapriya.R	20
33	922116104051	Vijayalakshmi.R	22
34	922116104052	Vimala Karthika.P	20
35	922116104054	Vishnu priya.V	21



Dr.D.SENTHIL KUMARAN M.E. Ph.D (IUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002



v.junni
HOD/CSE
(Dr.v.junni)



SSM Institute of Engineering and Technology, Dindigul-02

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

Feedback Form

Topic	Value Added Course on "Solving Problems using Python Programming"
Conducted by	Ms.S.Bharathi, AP/CSE Ms.V.Nivedhitha, AP/CSE Ms.N.Padmapriya,AP/CSE
Target Group	III & IV CSE Students
Date / Venue/Timing	06-08-2018 to 24-08-2018 & 03-09-2018 to 14-09-2018 / CSE Lab – 2 / 04.30 p.m. to 05.30 p.m.

1. Content delivery by Resource person

- Excellent
- Good
- Fair

2. Resource person's knowledge on content delivered

- Excellent
- Good
- Fair

3. Response of the resource person to students' queries.

- Excellent
- Good
- Fair

4. Whether the session was useful?

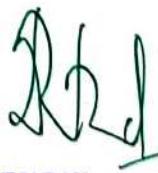
- Excellent
- Good
- Fair

5. Overall experience

- Excellent
- Good
- Fair

6. Suggestions/ Comments to improve, if any:




Dr. D. SENTHIL KUMARAN M.E., Ph.D (INUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002


Signature of the Participant



SSM Institute of Engineering and Technology, Dindigul-02
Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering
Feedback Form

Topic Value Added Course on "Solving Problems using Python Programming"
Conducted by Ms.S.Bharathi, AP/CSE
Ms.V.Nivedhitha, AP/CSE
Ms.N.Padmapriya,AP/CSE
Target Group III & IV CSE Students
Date / Venue/Timing 06-08-2018 to 24-08-2018 & 03-09-2018 to 14-09-2018 / CSE Lab – 2 /
04.30 p.m. to 05.30 p.m.

1. Content delivery by Resource person
 - Excellent
 - Good
 - Fair
2. Resource person's knowledge on content delivered
 - Excellent
 - Good
 - Fair
3. Response of the resource person to students' queries.
 - Excellent
 - Good
 - Fair
4. Whether the session was useful?
 - Excellent
 - Good
 - Fair
5. Overall experience
 - Excellent
 - Good
 - Fair
6. Suggestions/ Comments to improve, if any:



Dr.D.SENTHIL KUMARAN M.E.,Ph.D (NBS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palai Road, Dindigul - 624 102

B. Durga,
Signature of the Participant

Need more hours



SSM Institute of Engineering and Technology, Dindigul-02

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

Feedback Form

Topic	Value Added Course on "Solving Problems using Python Programming"
Conducted by	Ms.S.Bharathi, AP/CSE Ms.V.Nivedhitha, AP/CSE Ms.N.Padmapriya,AP/CSE
Target Group	III & IV CSE Students
Date / Venue/Timing	06-08-2018 to 24-08-2018 & 03-09-2018 to 14-09-2018 / CSE Lab – 2 / 04.30 p.m. to 05.30 p.m.

1. Content delivery by Resource person

- Excellent
- Good
- Fair

2. Resource person's knowledge on content delivered

- Excellent
- Good
- Fair

3. Response of the resource person to students' queries.

- Excellent
- Good
- Fair

4. Whether the session was useful?

- Excellent
- Good
- Fair

5. Overall experience

- Excellent
- Good
- Fair

6. Suggestions/ Comments to improve, if any:




Dr. D. SENTHIL KUMARAN M.E., Ph.D (INDS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002


Signature of the Participant



SSM Institute of Engineering and Technology, Dindigul-02

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

Feedback Form

Topic	Value Added Course on "Solving Problems using Python Programming"
Conducted by	Ms.S.Bharathi, AP/CSE Ms.V.Nivedhitha, AP/CSE Ms.N.Padmapriya,AP/CSE
Target Group	III & IV CSE Students
Date / Venue/Timing	06-08-2018 to 24-08-2018 & 03-09-2018 to 14-09-2018 / CSE Lab – 2 / 04.30 p.m. to 05.30 p.m.

1. Content delivery by Resource person

- Excellent
- Good
- Fair

2. Resource person's knowledge on content delivered

- Excellent
- Good
- Fair

3. Response of the resource person to students' queries.

- Excellent
- Good
- Fair

4. Whether the session was useful?

- Excellent
- Good
- Fair

5. Overall experience

- Excellent
- Good
- Fair

6. Suggestions/ Comments to improve, if any:



Dr.D.SENTHIL KUMARAN M.E.,Ph.D (INUS)
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

Signature of the Participant



SSM Institute of Engineering and Technology, Dindigul-02

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

Feedback Form

Topic	Value Added Course on "Solving Problems using Python Programming"
Conducted by	Ms.S.Bharathi, AP/CSE Ms.V.Nivedhitha, AP/CSE Ms.N.Padmapriya,AP/CSE
Target Group	III & IV CSE Students
Date / Venue/Timing	06-08-2018 to 24-08-2018 & 03-09-2018 to 14-09-2018 / CSE Lab – 2 / 04.30 p.m. to 05.30 p.m.

1. Content delivery by Resource person

- Excellent
- Good
- Fair

2. Resource person's knowledge on content delivered

- Excellent
- Good
- Fair

3. Response of the resource person to students' queries.

- Excellent
- Good
- Fair

4. Whether the session was useful?

- Excellent
- Good
- Fair

5. Overall experience

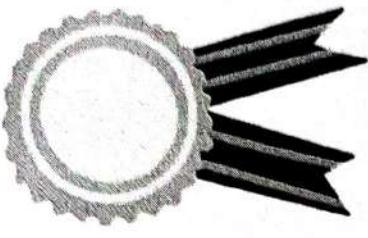
- Excellent
- Good
- Fair

6. Suggestions/ Comments to improve, if any:



Dr.D.SENTHIL KUMARAN M.E.P.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

Signature of the Participant



SSM
Institute of Engineering and Technology

CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS PRESENTED TO

Hari Prasath.M

Has successfully completed a course on "Solving Problem Using Python Programming"
organised by Department of Computer Science and Engineering



A.D.SENTHIL KUMARAN M.E. Ph.D IITB
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul - 624 002

V.Niveditha
PROGRAM COORDINATOR

Dr.V.Shanmugavel
HOD/CSE

DR.M.Saravanan
PRINCIPAL



SSM
Institute of Engineering and Technology

CERTIFICATE

OF COMPLETION

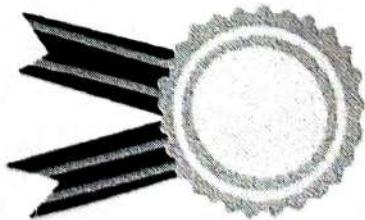
THIS CERTIFICATE IS PRESENTED TO

Durga.B

Has successfully completed a course on "Solving Problem Using Python Programming" organised by Department of Computer Science and Engineering

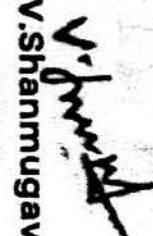



Dr. D. SENTHIL KUMARAN M.E., Ph.D., INDS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palai Road, Dindigul-624 002




V.Nivedhitha

PROGRAM COORDINATOR


Dr.V.Shanmugavel

HOD/CSE


DR.M.Saravanan

PRINCIPAL



SSM
Institute of Engineering and Technology

CERTIFICATE

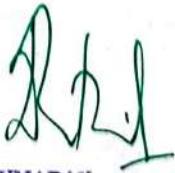
OF COMPLETION

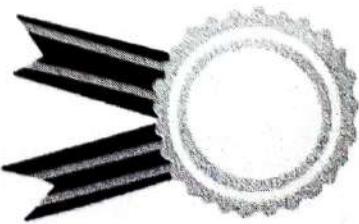
THIS CERTIFICATE IS PRESENTED TO

Shalini.R

Has successfully completed a course on "Solving Problem Using Python Programming"
organised by Department of Computer Science and Engineering



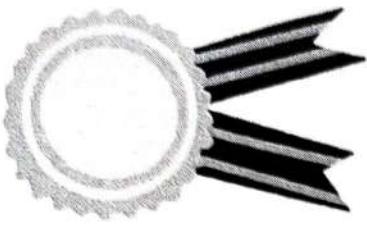

Dr. D. SENTHIL KUMARAN M.E., Ph.D., INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering and Technology
Palani Road, Dindigul-624 023




V.Nivedhitha
PROGRAM COORDINATOR


Dr. V. Shannugavel
HOD/CSE


DR.M.Saravanan
PRINCIPAL



SSM
Institute of Engineering and Technology

CERTIFICATE

OF COMPLETION

THIS CERTIFICATE IS PRESENTED TO

Abirami.S

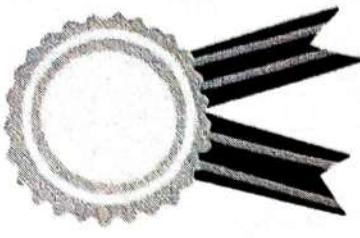
Has successfully completed a course on "Solving Problem Using Python Programming"
organised by Department of Computer Science and Engineering

Dr.D.SENTHIL KUMARAN M.E., Ph.D INUS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

V.Nivedhitha
PROGRAM COORDINATOR

HOD/CSE

DR.M.Saravanan
PRINCIPAL



SSM

Institute of Engineering and Technology



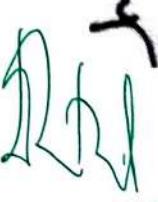
CERTIFICATE

OF COMPLETION

THIS CERTIFICATE IS PRESENTED TO

Ashwini.A

Has successfully completed a course on "Solving Problem Using Python Programming"
organised by Department of Computer Science and Engineering



V.Niveditha
PROGRAM COORDINATOR



DR.M.Saravanan
PRINCIPAL



Dr.V.Shanmugavel
HOD/CSE

Dr.D.SENTHIL SELVAMAN M.E., Ph.D, INBS
Senior Professor
Department of Mechanical Engineering
SSM Institute of Engineering Technology
Palani Road, Dindigul-624 002

