J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

6. CONTROL STRUCTURES

Section – A

Choose the best answer **(1 Mark)**

1. How many important control structures are there in Python?

A) 3 B) 4 C) 5 D) 6

2. elif can be considered to be abbreviation of

A) nested if B) if..else **C) else if** D) if..elif

3. What plays a vital role in Python programming?

A) Statements B) Control C) Structure **D) Indentation**

4. Which statement is generally used as a placeholder?

A) continue B) break **C) pass** D) goto

5. The condition in the if statement should be in the form of

A) Arithmetic or Relational expression B) Arithmetic or Logical expression

C) Relational or Logical expression D) Arithmetic

6. Which is the most comfortable loop?

A) do..while B) while **C) for** D) if..elif

7. What is the output of the following snippet?

```
i=1
while True:
if i%3 ==0:
break
print(i,end='')
i +=1
```

A) 1 2 B) 123 C) 1234 D) 124

8. What is the output of the following snippet?

```
T=1
while T:
print(True)
break
```

A) False **B) True** C) 0 D) no output

9. Which amongst this is not a jump statement ?

A) for B) goto C) continue D) break

10. Which punctuation should be used in the blank?

if <condition>_

statements-block 1

else:

statements-block 2

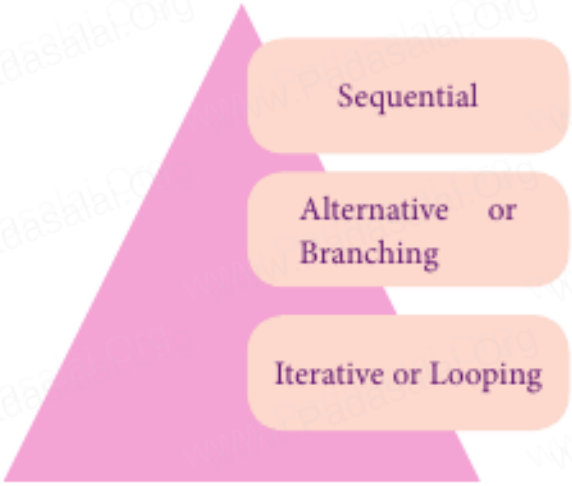
A) ; **B) :** C) :: D) !

Section-B

Answer the following questions **(2 Marks)**

1. List the control structures in Python.

- Three important control structures are,



2. Write note on break statement.

break statement :

- The **break** statement terminates the loop containing it.
- Control of the program flows to the statement immediately after the body of the loop.

3. Write is the syntax of if..else statement

Syntax:

if <condition>:

statements-block 1

else:

statements-block 2

4. Define control structure.

- A program statement that causes a jump of control from one part of the program to another is called control structure or control statement.

5. Write note on range () in loop

- range() generates a list of values starting from start till stop-1 in for loop.
- The syntax of range() is as follows:

range (start,stop,[step])

Where,

start – refers to the initial value

stop – refers to the final value

step – refers to increment value, this is optional part.

Section-C

Answer the following questions **(3 Marks)**

1. Write a program to display

A

A B

A B C

A B C D

A B C D E

CODE:

```
a=['A','B','C','D','E']
for i in range(0,6):
        for j in range(0,i):
                print(a[j],end=" ")
        else:
                print()
```

2. Write note on if..else structure.

- The **if .. else** statement provides control to check the true block as well as the false block.
- **if..else** statement thus provides two possibilities and the condition determines which BLOCK is to be executed.

Syntax:

```
if <condition>:
    statements-block 1
else:
    statements-block 2
```

3. Using if..else..elif statement write a suitable program to display largest of 3 numbers.

CODE:

```
n1= int(input("Enter the first number:"))
n2= int(input("Enter the first number:"))
n3= int(input("Enter the first number:"))
if(n1>=n2)and(n1>=n3):
   biggest=n1;
elif(n2>=n1)and(n2>=n3):
   biggest=n2
else:
   biggest=n3
print("The biggest number between",n1,",",n2,"and",n3,"is",biggest)
```

OUTPUT

```
Enter the first number:1
Enter the first number:3
Enter the first number:5
The biggest number between 1 , 3 and 5 is 5
```

4. Write the syntax of while loop.

Syntax:

```
while <condition>:
    statements block 1
[else:
    statements block2]
```

5. List the differences between break and continue statements.

break	continue
The **break** statement terminates the loop containing it.	The **Continue** statement is used to skip the remaining part of a loop and
Control of the program flows to the statement immediately after the body of the loop.	Control of the program flows start with next iteration.
Syntax: break	**Syntax:** continue

Section - D

Answer the following questions: **(5 Marks)**

1. Write a detail note on for loop.

- **for** loop is the most comfortable loop.
- It is also an entry check loop.
- The condition is checked in the beginning and the body of the loop(statements-block 1) is executed if it is only True otherwise the loop is not executed.

Syntax:

```
for counter_variable in sequence:
    statements-block 1
[else:      # optional block
    statements-block 2]
```

- The *counter_variable* is the control variable.
- The *sequence* refers to the initial, final and increment value.
- **for** loop uses the *range()* function in the sequence to specify the initial, final and increment values.
- **range()** generates a list of values starting from **start** till **stop-1.**

The syntax of range() is as follows:

range (start,stop,[step])

Where,

start – refers to the initial value

stop – refers to the final value

step – refers to increment value, this is optional part.

Example:

```
for i in range(2,10,2):
print (i,end=' ')
else:
print ("\nEnd of the loop")
```

Output:

```
2 4 6 8
End of the loop
```

2. Write a detail note on if..else..elif statement with suitable example.

Nested if..elif...else statement:

- When we need to construct a chain of **if** statement(s) then **'elif'** clause can be used instead of **'else'.**
- **'elif'** clause combines **if..else-if..else** statements to one **if..elif…else.**
- **'elif'** can be considered to be abbreviation of **'else if'.**
- In an **'if'** statement there is no limit of **'elif'** clause that can be used, but an **'else'** clause if used should be placed at the end.

Syntax:

```
if <condition-1>:
    statements-block 1
elif <condition-2>:
    statements-block 2
else:
    statements-block n
```

- In the syntax of **if..elif..else** mentioned above, condition-1 is tested if it is true then statements-block1 is executed.
- Otherwise the control checks condition-2, if it is true statements-block2 is executed and even if it fails statements-block n mentioned in **else** part is executed.

Example:

```
m1=int (input("Enter mark in first subject : "))
m2=int (input("Enter mark in second subject : "))
```

```
avg= (m1+m2)/2
if avg>=80:
        print (“Grade : A”)
elif avg>=70 and avg<80:
        print (“Grade : B”)
elif avg>=60 and avg<70:
        print (“Grade : C”)
elif avg>=50 and avg<60:
        print (“Grade : D”)

else:
        print (“Grade : E”)
```

Output :

Enter mark in first subject : 34

Enter mark in second subject : 78

Grade : D

3. Write a program to display all 3 digit odd numbers.

CODE:

```
lower=int(input("Enter the lower limit for the range:"))
upper=int(input("Enter the upper limit for the range:"))
for i in range(lower,upper+1):
   if(i%2!=0):
      print(i,end=" ")
```

Output:

```
Enter the lower limit for the range:100
Enter the upper limit for the range:150
101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143 145 147 149
>>>
```

4. Write a program to display multiplication table for a given number.

CODE:

```
num=int(input("Display Multiplication Table of "))
for i in range(1,11):
    print(i, 'x' ,num, '=' , num*i)
```

Output:

```
Display Multiplication Table of 2
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20
>>>
```

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

7. PYTHON FUNCTIONS

Section – A

Choose the best answer **(1 Mark)**

1. A named blocks of code that are designed to do one specific job is called as
   (a) Loop (b) Branching **(c) Function** (d) Block
2. A Function which calls itself is called as
   (a) Built-in **(b) Recursion** (c) Lambda (d) return
3. Which function is called anonymous un-named function
   (a) Lambda (b) Recursion (c) Function (d) define
4. Which of the following keyword is used to begin the function block?
   (a) define (b) for (c) finally **(d) def**
5. Which of the following keyword is used to exit a function block?
   (a) define **(b) return** (c) finally (d) def
6. While defining a function which of the following symbol is used.
   (a) ; (semicolon) (b) . (dot) **(c) : (colon)** (d) $ (dollar)
7. In which arguments the correct positional order is passed to a function?
   (a) Required (b) Keyword (c) Default (d) Variable-length
8. Read the following statement and choose the correct statement(s).
   (I) In Python, you don't have to mention the specific data types while defining function.
   (II) Python keywords can be used as function name.
   (a) I is correct and II is wrong
   (b) Both are correct
   (c) I is wrong and II is correct
   (d) Both are wrong
9. Pick the correct one to execute the given statement successfully.
   if ____ : print(x, " is a leap year")
   (a) x%2=0 **(b) x%4==0** (c) x/4=0 (d) x%4=0
10. Which of the following keyword is used to define the function testpython(): ?
    (a) define (b) pass **(c) def** (d) while

Section-B

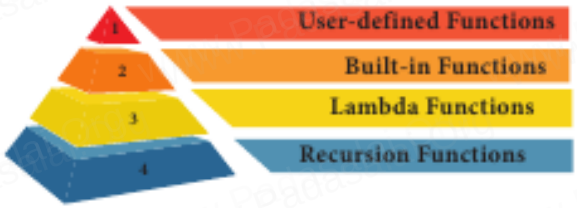
Answer the following questions **(2 Marks)**

1. What is function?

- Functions are named blocks of code that are designed to do one specific job.
- Types of Functions are User defined, Built-in, lambda and recursion.
- Function blocks begin with the keyword "def " followed by function name and parenthesis ().

2. Write the different types of function.

TYPES OF FUNCTION:



3. What are the main advantages of function?

- **Main advantages of functions are ,**
  - It avoids repetition and makes high degree of code reusing.
  - It provides better modularity for your application.

4. What is meant by scope of variable? Mention its types.

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- Scope holds the current set of variables and their values.
- The two types of scopes are- **local scope** and **global scope**.

5. Define global scope.

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

6. What is base condition in recursive function

- A recursive function calls itself.
- The condition that is applied in any recursive function is known as base condition.
- A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

7. How to set the limit for recursive function? Give an example.

- Python stops calling recursive function after 1000 calls by default.
- So, It also allows you to change the limit using sys.setrecursionlimit (limit_value).
- **Example:**

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
if n == 0:
return 1
else:
return n * fact(n-1)
print(fact (2000))
```

Section-C

Answer the following questions **(3 Marks)**

1. Write the rules of local variable.

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing.
- The formal arguments are also local to function.

2. Write the basic rules for global keyword in python.

The basic rules for ***global*** keyword in Python are:

- When we define a variable outside a function, it's global by default. You don't have to use global keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect.

3. What happens when we modify global variable inside the function?

- If we modify the global variable , We can see the change on the **global** variable outside the function also.

Example:

```
x = 0                                          # global variable
def add():
    global x
    x = x + 5                                  # increment by 2
    print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

Output:

```
Inside add() function x value is : 5
In main x value is : 5                         #value of x changed outside the function
```

4. Differentiate ceil() and floor() function?

ceil()	floor()
Returns the smallest integer greater than or equal to x	Returns the largest integer less than or equal to x
math.ceil (x)	math.floor (x)

5. Write a Python code to check whether a given year is leap year or not.

CODE:

```
n=int(input("Enter the year"))
if(n%4==0):
   print ("Leap Year")
else:
   print ("Not a Leap Year")
```

Output:

```
Enter the year      2012
Leap Year
```

6. What is composition in functions?

- The value returned by a function may be used as an argument for another function in a nested manner.
- This is called **composition**.
- **For example,** if we wish to take a numeric value as a input from the user, we take the input string from the user using the function **input()** and apply **eval()** function to evaluate its value.

7. How recursive function works?

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

8. What are the points to be noted while defining a function?

When defining functions there are multiple things that need to be noted;

- Function blocks begin with the keyword **"def"** followed by function name and parenthesis ().
- Any input parameters should be placed within these parentheses.
- The code block always comes after a colon (:) and is indented.
- The statement **"return [expression]"** exits a function, and it is optional.
- A **"return"** with no arguments is the same as return None.

Section - D

Answer the following questions: **(5 Marks)**

1. Explain the different types of function with an example.

- Functions are named blocks of code that are designed to do one specific job.
- **Types of Functions**
  - User defined Function
  - Built-in Function
  - Lambda Function
  - Recursion Function

i) BUILT-IN FUNCTION:

- Built-in functions are Functions that are inbuilt with in Python.
- print(), echo() are some built-in function.

ii) USER DEFINED FUNCTION:

- Functions defined by the users themselves are called user defined function.
- Functions must be defined, to create and use certain functionality.
- Function blocks begin with the keyword "def " followed by function name and parenthesis ().
- When defining functions there are multiple things that need to be noted;
  - Function blocks begin with the keyword **"def"** followed by function name and parenthesis ().
  - Any input parameters should be placed within these parentheses.
  - The code block always comes after a colon (:) and is indented.
  - The statement **"return [expression]"** exits a function, and it is optional.
  - A **"return"** with no arguments is the same as return None.
- **EXAMPLE:**

```
def area(w,h):
    return w * h
print (area (3,5))
```

iii) LAMBDA FUNCTION:

- In Python, anonymous function is a function that is defined without a name.
- While normal functions are defined using the **def** keyword, in Python anonymous functions are defined using the **lambda** keyword.
- Hence, anonymous functions are also called as **lambda** functions.

USE OF LAMBDA OR ANONYMOUS FUNCTION:

- Lambda function is mostly used for creating small and one-time anonymous function.
- Lambda functions are mainly used in combination with the functions like filter(), map() and reduce().

EXAMPLE:

```
sum = lambda arg1, arg2: arg1 + arg2
print ('The Sum is :', sum(30,40))
print ('The Sum is :', sum(-30,40))
```

Output:

```
The Sum is : 70
The Sum is : 10
```

iv) RECURSIVE FUNCTION:

Functions that calls itself is known as recursive.

Overview of how recursive function works

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

2. Explain the scope of variables with an example.

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- We can say that scope holds the current set of variables and their values.
- There are two types of scopes - **local scope** and **global scope**.

➢ **Local Scope:**

- A variable declared inside the function's body or in the local scope is known as local variable.

Rules of local variable:

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing.
- The formal arguments are also local to function.

Example:

```
def loc():
y=0 # local scope
print(y)
loc()
```

Output:

```
0
```

➢ **Global Scope**

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

➢ **Rules of global Keyword**

The basic rules for ***global*** keyword in Python are:

- When we define a variable outside a function, it's global by default. You don't have to use global keyword.

- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect

Use of global Keyword

- Without using the global keyword we cannot modify the global variable inside the function but we can only access the global variable.

Example:

```
x = 0                                   # global variable
def add():
global x
x = x + 5                               # increment by 2
print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

Output:

```
Inside add() function x value is : 5
In main x value is : 5                  #value of x changed outside the function
```

3. Explain the following built-in functions.

(a) id() (b) chr() (c) round() (d) type() (e) pow()

Function	Description	Syntax	Example
id ( )	Return the "identity" of an object. i.e. the address of the object in memory.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **Output:** address of x is : 1357486752 address of y is : 13480736
chr ( )	Returns the Unicode character for the given ASCII value.	chr (i)	c=65 print (chr (c)) **Output:** A
round ( )	Returns the nearest integer to its input. 1. First argument (number) is used to specify the value to be rounded.	round (number [,ndigits])	x= 17.9 print ('x value is rounded to', round (x)) **Output:** X value is rounded to 18

type ( )	Returns the type of object for the given single object.	type (object)	x= 15.2 print (type (x)) **Output:** <class 'float'>
pow ( )	Returns the computation of a,b i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 b= 2 print (pow (a,b)) **Output:** 25

4. Write a Python code to find the L.C.M. of two numbers.

CODE:

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
if x>y:
    min=x
else:
    min=y

while(1):
  if((min%x == 0) and (min % y == 0)):
      print("LCM is:",min)
      break
  min=min+1
```

OUTPUT:

```
Enter first number:2
Enter second number:3
LCM is: 6
```

5. Explain recursive function with an example.

- Functions that calls itself is known as recursive.
- When a function calls itself is known as recursion.
- Recursion works like loop but sometimes it makes more sense to use recursion than loop.

- Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.
- The condition that is applied in any recursive function is known as base condition.
- A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.
- Python stops calling recursive function after 1000 calls by default.
- So, It also allows you to change the limit using sys.setrecursionlimit (limit_value).

Overview of how recursive function works:

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

EXAMPLE:

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact (n-1)
    print (fact (0))
    print (fact (5))
```

Output:

1
120

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

8. STRINGS AND STRING MANIPULATION

Section – A

Choose the best answer **(1 Mark)**

1. Which of the following is the output of the following python code?

```
str1="TamilNadu"
print(str1[::-1])
```

(a) Tamilnadu (b) Tmlau (c) udanlimaT **d) udaNlimaT**

2. What will be the output of the following code?

```
str1 = "Chennai Schools"
str1[7] = "-"
```

(a) Chennai-Schools (b) Chenna-School **(c) Type error** (d) Chennai

3. Which of the following operator is used for concatenation?

(a) + (b) & (c) * (d) =

4. Defining strings within triple quotes allows creating:

(a) Single line Strings **(b) Multiline Strings**
(c) Double line Strings (d) Multiple Strings

5. Strings in python:

(a) Changeable (b) Mutable **(c) Immutable** (d) flexible

6. Which of the following is the slicing operator?

(a) { } **(b) [ ]** (c) < > (d) ( )

7. What is stride?

(a) index value of slide operation (b) first argument of slice operation
(c) second argument of slice operation **(d) third argument of slice operation**

8. Which of the following formatting character is used to print exponential notation in upper case?

(a) %e **(b) %E** (c) %g (d) %n

9. Which of the following is used as placeholders or replacement fields which get replaced along with format( ) function?

(a) { } (b) < > (c) ++ (d) ^^

10. The subscript of a string may be:

(a) Positive (b) Negative (c) Both (a) and (b) **(d) Either (a) or (b)**

Section-B

Answer the following questions **(2 Marks)**

1. What is String?

- String is a data type in python, used to handle array of characters.
- String is a sequence of characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

2. Do you modify a string in Python?

- Yes we can modify the string by the following method,
- A new string value can be assign to the existing string variable.
- When defining a new string value to the existing string variable.
- Python completely overwrite new string on the existing string.

3. How will you delete a string in Python?

- Python will not allow deleting a particular character in a string.
- Whereas you can remove entire string variable using **del** command.
- **Example:**

```
del str1[2]
```

4. What will be the output of the following python code?

```
str1 = "School"
print(str1*3)
```

OUTPUT:

School School School

5. What is slicing?

- Slice is a substring of a main string.
- A substring can be taken from the original string by using [ ] slicing operator and index or subscript values.
- Using slice operator, you have to slice one or more substrings from a main string.

General format of slice operation:

str[start:end]

Section-C

Answer the following questions **(3 Marks)**

1. Write a Python program to display the given pattern

C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
C O M
C O
C

CODE:

```
str="COMPUTER"
index=len(str)
for i in str:
    print(str[:index])
    index-=1
```

2. Write a short about the followings with suitable example: (a) capitalize( ) (b) swapcase( )

FUNCTION	PURPOSE	EXAMPLE

capitalize( )	Used to capitalize the first character of the string	>>> city="chennai"
		>>> print(city.capitalize())
		Output:
		Chennai

swapcase( )	It will change case of every character to its opposite case vice-versa.	>>> str1="tAmiL NaDu"
		>>> print(str1.swapcase())
		Output:
		TaMIl nAdU

3. What will be the output of the given python program?

CODE:

```
str1 = "welcome"
str2 = "to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

OUTPUT:

weol

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/x.py =======
weol
>>>
```

4. What is the use of format( )? Give an example.

- The **format( )** function used with strings is very powerful function used for formatting strings.
- The curly braces { } are used as placeholders or replacement fields which get replaced along with format( ) function.

EXAMPLE:

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

OUTPUT:

```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88
```

5. Write a note about count( ) function in python.

- Returns the number of substrings occurs within the given range.
- Remember that substring may be a single character.
- Range (beg and end) arguments are optional. If it is not given, python searched in whole string.
- Search is case sensitive.

SYNTAX:

count(str, beg, end)

EXAMPLE:

```
>>> str1="Raja Raja Chozhan"
>>> print(str1.count('Raja'))
```

OUTPUT: 2

Section - D

Answer the following questions: **(5 Marks)**

1. Explain about string operators in python with suitable example.

STRING OPERATORS

Python provides the following string operators to manipulate string.

(i) Concatenation (+)

- Joining of two or more strings using plus **(+) operator** is called as **Concatenation**.

Example

```
>>> "welcome" + "Python"
```

Output: **'welcomePython'**

(ii) Append (+ =)

- Adding more strings at the end of an existing string using **operator +=** is known as **append.**

Example:

```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
```

Output: ***Welcome to Learn Python***

(iii) Repeating (*)

- The multiplication operator (*) is used to display a string in multiple number of times.

Example:

```
>>> str1="Welcome "
>>> print (str1*4)
```

Output: Welcome Welcome Welcome Welcome

(iv) String slicing

- Slice is a substring of a main string.
- A substring can be taken from the original string by using **[ ] slicing operator** and index values.
- Using slice operator, you have to slice one or more substrings from a main string.

General format of slice operation:

str[start:end]

- Where ***start*** is the beginning index and ***end*** is the last index value of a character in the string.
- Python takes the end value less than one from the actual index specified.

Example: slice a single character from a string

```
>>> str1="THIRUKKURAL"
>>> print (str1[0])
```

Output: T

(v) Stride when slicing string

- When the slicing operation, you can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string.
- The default value of stride is 1.
- Python takes the last value as n-1
- You can also use negative value as stride, to prints data in reverse order.

Example:

```
>>> str1 = "Welcome to learn Python"
>>> print (str1[10:16])
>>> print(str1[::-2])
```

Output: Learn

nhy re teolW

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

9. LISTS, TUPLES, SETS, AND DICTIONARY

Section – A

Choose the best answer **(1 Mark)**

1. Pick odd one in connection with collection data type

(a) List (b) Tuple (c) Dictionary **(d) Loop**

2. Let list1=[2,4,6,8,10], then print(List1[-2]) will result in

(a) 10 **(b) 8** (c) 4 (d) 6

3. Which of the following function is used to count the number of elements in a list?

(a) count() (b) find() **(c)len()** (d) index()

4. If List=[10,20,30,40,50] then List[2]=35 will result

(a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35] **(c) [10,20,35,40,50]** (d) [10,35,30,40,50]

5. If List=[17,23,41,10] then List.append(32) will result

(a) [32,17,23,41,10] **(b) [17,23,41,10,32]** (c) [10,17,23,32,41] (d) [41,32,23,17,10]

6. Which of the following Python function can be used to add more than one element within an existing list?

(a) append() (b) append_more() **(c)extend()** (d) more()

7. What will be the result of the following Python code?

```
S=[x**2 for x in range(5)]
print(S)
```

(a) [0,1,2,4,5] (b) [0,1,4,9,16] (c) [0,1,4,9,16,25] **(d) [1,4,9,16,25]**

8. What is the use of type() function in python?

(a) To create a Tuple (b) To know the type of an element in tuple.

(c) To know the data type of python object. (d) To create a list.

9. Which of the following statement is not correct?

(a) A list is mutable

(b) A tuple is immutable.

(c) The append() function is used to add an element.

(d) The extend() function is used in tuple to add elements in a list.

10. Let setA={3,6,9}, setB={1,3,9}. What will be the result of the following snippet?

```
print(setA|setB)
```

(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} **(d) {1,3,6,9}**

11. Which of the following set operation includes all the elements that are in two sets but not the one that are common to two sets?

(a) Symmetric difference (b) Difference (c) Intersection (d) Union

12. The keys in Python, dictionary is specified by

(a) = (b) ; (c)+ **(d) :**

Section-B

Answer the following questions **(2 Marks)**

1. What is List in Python?

- A list is an ordered collection of values enclosed within square brackets [ ] also known as a "sequence data type".
- Each value of a list is called as element.
- Elements can be a numbers, characters, strings and even the nested lists.
- **Syntax:** Variable = [element-1, element-2, element-3 …… element-n]

2. How will you access the list elements in reverse order?

- Python enables reverse or negative indexing for the list elements.
- A negative index can be used to access an element in reverse order.
- Thus, python lists index in opposite order.
- The python sets -1 as the index value for the last element in list and -2 for the preceding element and so on.
- This is called as **Reverse Indexing.**

3. What will be the value of x in following python code?

```
List1=[2,4,6,[1,3,5]]
x=len(List1)
print(x)
```

OUTPUT:

```
====== RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/LI.py ======
4
>>>
```

4. Differentiate del with remove( ) function of List.

del	remove( )
del statement is used to delete known elements	**remove( )** function is used to delete elements of a list if its index is unknown.
The del statement can also be used to delete entire list.	The remove is used to delete a particular element

5. Write the syntax of creating a Tuple with n number of elements.

Syntax:

```
Tuple_Name = (E1, E2, E2 ……. En)   # Tuple with n number elements
Tuple_Name = E1, E2, E3 ….. En     # Elements of a tuple without parenthesis
```

6. What is set in Python?

- In python, a set is another type of collection data type.
- A Set is a mutable and an unordered collection of elements without duplicates or repeated element.
- This feature used to include membership testing and eliminating duplicate elements.

Section-C

Answer the following questions **(3 Marks)**

1. What are the advantages of Tuples over a list?

- The elements of a list are changeable (mutable) whereas the elements of a tuple are unchangeable (immutable), this is the key difference between tuples and list.
- The elements of a list are enclosed within square brackets. But, the elements of a tuple are enclosed by paranthesis.
- Iterating tuples is faster than list.

2. Write a short note about sort( ).

sort ( ):

- It sorts the element in list.
- sort( ) will affect the original list.

Syntax : List.sort(reverse=True|False, key=myFunc)

Description of the Syntax:

Both arguments are optional ,

- If reverse is set as True, list sorting is in descending order.
- Ascending is default.
- Key=myFunc; “myFunc” - the name of the user defined function that specifies the sorting criteria.

3. What will be the output of the following code?

```
list = [2**x for x in range(5)]
print(list)
```

OUTPUT: [1, 2, 4, 8, 16]

4. Explain the difference between del and clear( ) in dictionary with an example.

del	clear( )
The **del** statement is used to delete known elements	The function **clear( )** is used to delete all the elements in list
The del statement can also be used to delete entire list.	It deletes only the elements and retains the list.

5. List out the set operations supported by python.

Set Operations:

(i) Union: It includes all elements from two or more sets.

(ii) Intersection: It includes the common elements in two sets.

(iii) Difference: It includes all elements that are in first set (say set A) but not in the second set (say set B).

iv) Symmetric difference: It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.

6. What are the difference between List and Dictionary?

List	Dictionary
• A list is an ordered collection of values or elements of any type .	• A dictionary is a mixed collection of elements and it stores a key along with its element.
• It is enclosed within square brackets [ ]	• The key value pairs are enclosed with curly braces { }.
• **Syntax:** Variable = [element-1, element-2, element-3 …… element-n]	• **Syntax of defining a dictionary:** Dictionary_Name = { Key_1: Value_1, Key_2:Value_2, …….. Key_n:Value_n }
• The commas work as a separator for the elements.	• The keys in a Python dictionary is separated by a colon (:) while the commas work as a separator for the elements.

Section - D

Answer the following questions: **(5 Marks)**

1. What the different ways to insert an element in a list. Explain with suitable example.

Inserting elements in a list using insert():

- The **insert ( )** function helps you to include an element at your desired position.
- The **insert( )** function is used to insert an element at any position of a list.

Syntax:

List.insert (position index, element)

Example:

```
>>> MyList=[34,98,47,'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan' ]
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
```

Output: [34, 98, 47, 'Ramakrishnan', 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']

- In the above example, insert( ) function inserts a new element ‘Ramakrishnan’ at the index value 3, ie. at the 4^{th} position.
- While inserting a new element, the existing elements shifts one position to the right.

Adding more elements in a list using append():

- The **append( )** function is used to add a single element in a list.
- But, it includes elements at the end of a list.

Syntax:

List.append (element to be added)

Example:

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
```

Output: [34, 45, 48, 90]

Adding more elements in a list using extend():

- The **extend( )** function is used to add more than one element to an existing list.
- In extend( ) function, multiple elements should be specified within square bracket as arguments of the function.

Syntax:

List.extend ([elements to be added])

Example:

```
>>> Mylist=[34, 45, 48]
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
```

Output: [34, 45, 48, 90, 71, 32, 29]

2. What is the purpose of range( )? Explain with an example.

range():

- The range( ) is a function used to generate a series of values in Python.
- Using range( ) function, you can create list with series of values.
- The range( ) function has three arguments.

Syntax of range ( ) function:

range (start value, end value, step value)

where,

- **start value** – beginning value of series. Zero is the default beginning value.
- **end value** – upper limit of series. Python takes the ending value as upper limit – 1.
- **step value** – It is an optional argument, which is used to generate different interval of values.

Example : Generating whole numbers upto 10

```
for x in range (1, 11):
print(x)
```

Output:

1
2
3
4
5
6
7
8
9
10

Creating a list with series of values

- Using the range( ) function, you can create a list with series of values.
- To convert the result of range( ) function into list, we need one more function called list( ).
- The list( ) function makes the result of range( ) as a list.

Syntax:

List_Varibale = list (range ( ))

Example :

```
>>> Even_List = list(range(2,11,2))
>>> print(Even_List)
```

Output: [2, 4, 6, 8, 10]

- In the above code, list( ) function takes the result of range( ) as Even_List elements.
- Thus, Even_List list has the elements of first five even numbers.

3. What is nested tuple? Explain with an example.

Tuple:

- Tuples consists of a number of values separated by comma and enclosed within parentheses.
- Tuple is similar to list, values in a list can be changed but not in a tuple.

Nested Tuples:

- In Python, a tuple can be defined inside another tuple; called Nested tuple.
- In a nested tuple, each tuple is considered as an element.
- The for loop will be useful to access all the elements in a nested tuple.

Example:

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5), ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
for i in Toppers:
    print(i)
```

Output:
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)

4. Explain the different set operations supported by python with suitable example.

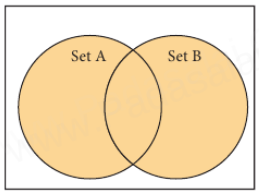
- A Set is a mutable and an unordered collection of elements without duplicates.

Set Operations:

- The set operations such as Union, Intersection, difference and Symmetric difference.

(i) Union:

- It includes all elements from two or more sets.
- The **operator** | is used to union of two sets.
- The function union( ) is also used to join two sets in python.



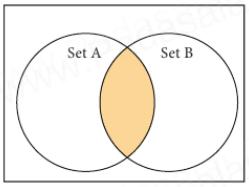
Example:

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

Output:
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}

(ii) Intersection:

- It includes the common elements in two sets.
- The **operator &** is used to intersect two sets in python.
- The function **intersection( )** is also used to intersect two sets in python.



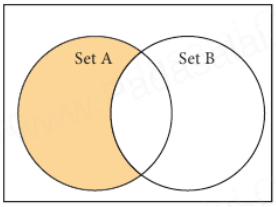
Example:

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

Output:
{'A', 'D'}

(iii) Difference:

- It includes all elements that are in first set (say set A) but not in the second set (say set B).
- The minus **(-) operator** is used to difference set operation in python.
- The function **difference( )** is also used to difference operation.



Example:

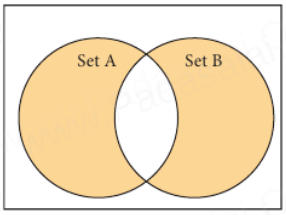
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

Output:

{2, 4}

(iv) Symmetric difference

- It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.
- The caret **(^) operator** is used to symmetric difference set operation in python.
- The function **symmetric_difference( )** is also used to do the same operation.



Example:

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```

Output:

{2, 4, 'B', 'C'}

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

11. DATABASE CONCEPTS

Section – A

Choose the best answer **(1 Mark)**

1. What is the acronym of DBMS?
   a) DataBase Management Symbol b) Database Managing System
   c) DataBase Management System d) DataBasic Management System
2. A table is known as
   a) tuple b) attribute **c) relation** d)entity
3. Which database model represents parent-child relationship?
   a) Relational b) Network **c) Hierarchical** d) Object
4. Relational database model was first proposed by
   a) E F Codd b) E E Codd c) E F Cadd d) E F Codder
5. What type of relationship does hierarchical model represents?
   a) one-to-one **b) one-to-many** c) many-to-one d) many-to-many
6. Who is called Father of Relational Database from the following?
   a) Chris Date b)Hugh Darween **c) Edgar Frank Codd** d) Edgar Frank Cadd
7. Which of the following is an RDBMS?
   a) Dbase b) Foxpro c) Microsoft Access **d) SQLite**
8. What symbol is used for SELECT statement?
   a) σ b) Π c) X d) Ω
9. A tuple is also known as
   a) table **b) row** c) attribute d) field
10. Who developed ER model?
    a) Chen b) EF Codd c) Chend d) Chand

Section-B

Answer the following questions **(2 Marks)**

1. Mention few examples of a database.

- Foxpro
- dbase.
- IBM DB2.
- Microsoft Access.
- Microsoft Excel.
- MySQL.

2. List some examples of RDBMS.

- SQL Server
- Oracle
- MySQL
- MariaDB
- SQLite

3. What is data consistency?

- Data Consistency means that data values are the same at all instances of a database.
- On live data, it is being continuously updated and added, maintaining the consistency of data can become a challenge.
- But DBMS handles it by itself.

4. What is the difference between Hierarchical and Network data model?

Hierarchical data model	Network data model
• In hierarchical model, a child record has only one parent node	• In a Network model, a child may have many parent nodes.
• It represents one-to-one relationship called parent-child relationship in the form of tree structure.	• It represents the data in many-to-many relationships.

5. What is normalization?

- Normalization is an integral part of RDBMS in order to reduce data redundancy and improve data integrity.

Section-C

Answer the following questions **(3 Marks)**

1. What is the difference between Select and Project command?

Select Command	Project Command
• The SELECT operation is used for selecting a subset with tuples according to a given condition C.	• The projection method defines a relation that contains a vertical subset of Relation.
• Select filters out all tuples that do not satisfy C.	• The projection eliminates all attributes of the input relation but those mentioned in the projection list.
Symbol : σ	**Symbol :** Π
General Form: $\sigma_c(R)$ **Example:** $\sigma_{course = \text{“Big Data”}}(STUDENT)$	**Example:** $\Pi_{course}(STUDENT)$

2. What is the role of DBA?

- Database Administrator or DBA is the one **who manages the complete database** management system.
- DBA takes care of the security of the DBMS, managing the license keys, managing user accounts and access etc.

3. Explain Cartesian Product with a suitable example.

- Cross product is a way of combining two relations.
- The resulting relation contains, both relations being combined.
- This type of operation is helpful to merge columns from two relations.
- **Example:** A x B means A times B, where the relation A and B have different attributes.

4. Explain Object Model with example.

- Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.
- It is used in file Management System.
- It represents real world objects, attributes and behaviors.

5. Write a note on different types of DBMS users.

Database Administrators

- Database Administrator or DBA is the one who manages the complete database management system.

Application Programmers or Software Developers

- This user group is involved in developing and designing the parts of DBMS.

End User

- End users are the one who store, retrieve, update and delete data.

Database designers:

- They are responsible for identifying the data to be stored in the database for choosing appropriate structures to represent and store the data.

Section - D

Answer the following questions: **(5 Marks)**

1. Explain the different types of data model.

Data Model

A data model describes how the data can be represented and accessed from a software after complete implementation

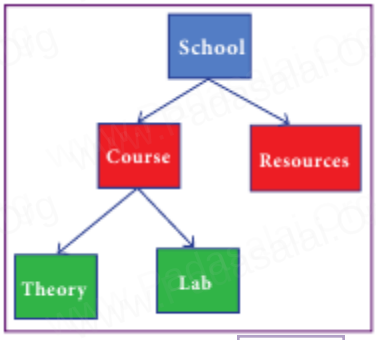
Types of Data Model

The different types of a Data Model are,

- Hierarchical Model
- Relational Model
- Network Database Model
- Entity Relationship Model
- Object Model

i). Hierarchical Model:

- In Hierarchical model, data is represented as a simple tree like structure form.
- This model represents a one-to-many relationship ie parent-child relationship.
- One child can have only one parent but one parent can have many children.
- This model is mainly used in IBM Main Frame computers.
- **Example:**



Hierarchical Model

ii). Relational Model

- The Relational Database model was first proposed by E.F. Codd in 1970 .
- The basic structure of data in relational model is tables (relations).
- All the information's related to a particular type is stored in rows of that table.
- Hence tables are also known as relations in a relational model.
- A relation key is an attribute which uniquely identifies a particular tuple (row in a relation (table)).
- **Example:**

Stu_id	Name	Age
1	Malar	17
2	Suncar	16
3	Velu	16

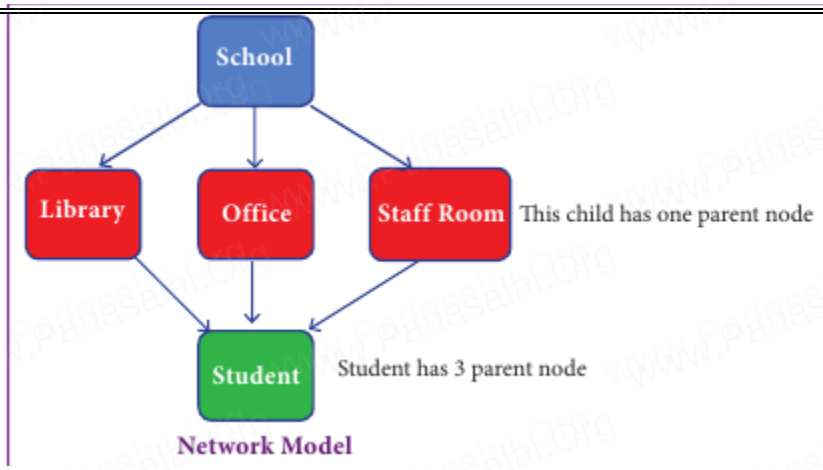
Subj_id	Name	Teacher
1	C++	Kannan
2	Php	Ramakrishnan
3	Python	Vidhya

Stu_id	Subj_id	Marks
1	1	92
1	2	89
3	2	96

Relational Model

iii.) Network Model

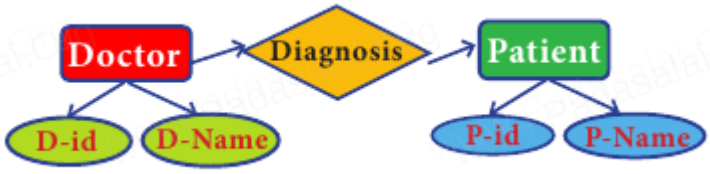
- Network database model is an extended form of hierarchical data model.
- In a Network model, a child may have many parent nodes.
- It represents the data in many-to-many relationships.
- This model is easier and faster to access the data.



Network Model

iv.) Entity Relationship Model. (ER model)

- In this database model, relationship are created by dividing the object into entity and its characteristics into attributes.
- It was developed by Chen in 1976.
- ER model constructed by,
  - **Rectangle** represents the entities.
  - **Ellipse** represents the attributes .
  - **Attributes** describes the characteristics and each entity.
  - **Diamond** represents the relationship in ER diagrams
  - **Example: Doctor diagnosis the Patient.**

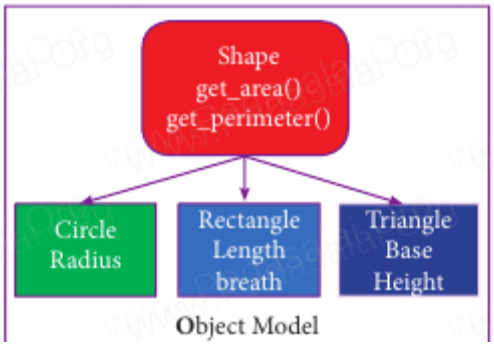


ER model

v.) Object Model

- Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.

Example:



Object Model

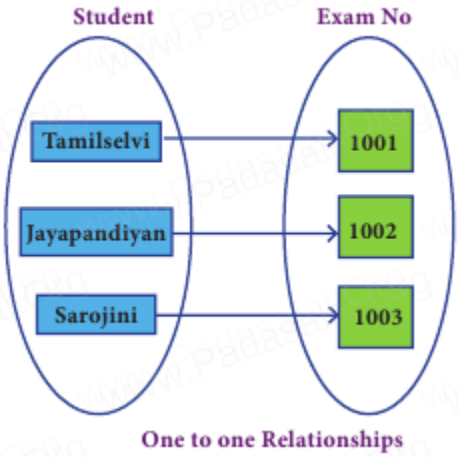
2. Explain the different types of relationship mapping.

Types of Relationships : There are the types of relationships used in a database.

1. One-to-One Relationship
2. One-to-Many Relationship
3. Many-to-One Relationship
4. Many-to-Many Relationship

i.) One-to-One Relationship:

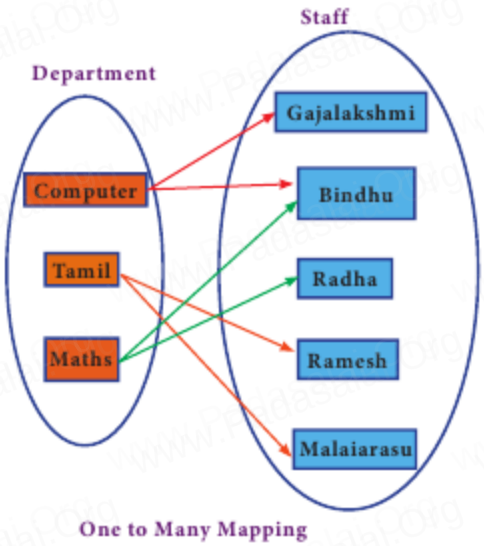
- In One-to-One Relationship, one entity is related with only one other entity.
- One row in a table is linked with only one row in another table and vice versa.
- **For Example:** A student can have only one exam number.



One to one Relationships

ii. One-to-Many Relationship:

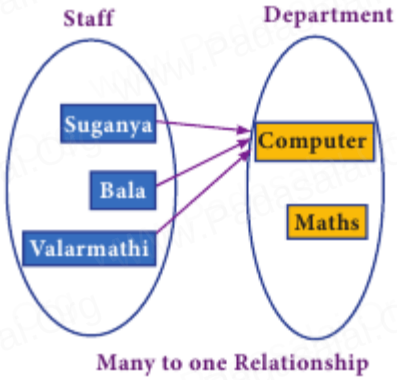
- In One-to-Many relationship, one entity is related to many other entities.
- One row in a table A is linked to many rows in a table B, but one row in a table B is linked to only one row in table A.
- **For Example:** One Department has many staff members.



One to Many Mapping

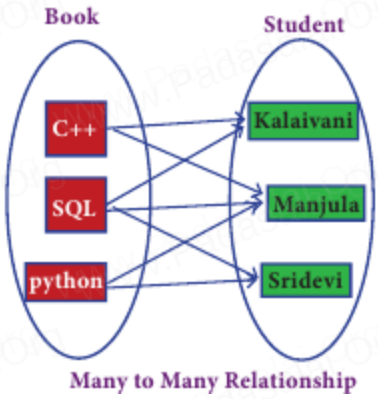
iii. Many-to-One Relationship:

- In Many-to-One Relationship, many entities can be related with only one in the other entity.
- **For Example:** A number of staff members working in one Department.
- Multiple rows in staff members table is related with only one row in Department table.



4. Many-to-Many Relationship:

- A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table.
- **Example: Books and Student :**Many Books in a Library are issued to many students.



3. Differentiate DBMS and RDBMS.

Basis of Comparison	DBMS	RDBMS
Expansion	Database Management System	Relational DataBase Management System
Data storage	Navigational model ie data by linked records	Relational model (in tables). ie data in tables as row and column
Data redundancy	Exhibit	Not Present
Normalization	Not performed	RDBMS uses normalization to reduce redundancy

Data access	Consumes more time	Faster, compared to DBMS.
Keys and indexes	Does not use.	Used to establish relationship. Keys are used in RDBMS.
Transaction management	Inefficient, Error prone and insecure.	Efficient and secure.
Distributed Databases	Not supported	Supported by RDBMS.
Example	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite.

4. Explain the different operators in Relational algebra with suitable examples.

- Relational Algebra is used for modeling data stored in relational databases and for defining queries on it.
- Relational Algebra is divided into various groups.

1) Unary Relational Operations

- SELECT (symbol : σ)
- PROJECT (symbol : Π)

2) Relational Algebra Operations from Set Theory

- UNION (∪)
- INTERSECTION (∩)
- DIFFERENCE (−)
- CARTESIAN PRODUCT (X)

- **SELECT (symbol : σ)**

- General form σ_{c} (R) with a relation R and a condition C on the attributes of R.
- The SELECT operation is used for selecting a subset with tuples according to a given condition.
- Select filters out all tuples that do not satisfy C.
- **Example:** σ_{course} = "Big Data" **(STUDENT)**

- **PROJECT (symbol : Π)**

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
  - The projection method defines a relation that contains a vertical subset of Relation.

- **Example:** Π_{course} (STUDENT)

- **UNION (Symbol :∪)** **A** ∪ **B**

- It includes all tuples that are in tables A or in B.
- It also eliminates duplicates.
- Set A Union Set B would be expressed as A ∪ B

➢ **SET DIFFERENCE (Symbol : -)**

- The result of A – B, is a relation which includes all tuples that are in A but not in B.
- The attribute name of A has to match with the attribute name in B.

➢ **INTERSECTION (symbol : ∩) A ∩ B**

- Defines a relation consisting of a set of all tuple that are in both in A and B.
- However, A and B must be union-compatible.

➢ **PRODUCT OR CARTESIAN PRODUCT (Symbol : X)**

- Cross product is a way of combining two relations.
- The resulting relation contains, both relations being combined.
- This type of operation is helpful to merge columns from two relations.
- A x B means A times B, where the relation A and B have different attributes.

5. Explain the characteristics of DBMS.

1. Data Stored in a Tables	• Data is stored into tables, created inside the database. • DBMS also allows to have relationship between tables.
2. Reduced Redundancy	• Unnecessary repetition of data in database was a big problem. • DBMS follows Normalisation which divides the data in such a way that repetition is minimum.
3.Data Consistency	• Data Consistency means that data values are the same at all instances of a database.
4.Support Multiple user and Concurrent Access	• DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
5.Query Language	• DBMS provides users with a simple query language, using which data can be easily fetched, inserted, deleted and updated in a database.
6. Security	• The DBMS also takes care of the security of data, protecting the data from unauthorized access. • Creating user accounts with different access permissions we can easily secure our data.
7. DBMS Supports Transactions	• It allows us to better handle and manage data integrity in real world applications where multi-threading is extensively used.

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com