N.S.Hr.Sec.School, Theni

XII – Computer Science (5M Questions)

Lesson 1 - Function

1) What are called Parameters and write a note on (i) Parameter without Type (ii) Parameter with Type

Parameters are the variables in a function definition and arguments are the values which are passed to a function definition.

Parameter without Type

Let us see an example of a function definition:

```
let bigger a  b  :=
        if a > b then return a
        else return b
```

* Variables a and b are parameters.
* We have not mentioned any data types.
* Some language compilers solve this type inference algorithmically.

Parameter with Type

Now let us write the same function definition with types.

```
let bigger (a:int)  (b:int)  :int  :=
        if a > b then return a
        else return b
```

* Here we have mentioned the data types.
* Parentheses are mandatory to mention the data type for a and b.
* This is useful on times when you get a type error from the compiler.

2. Identify in the following program

```
let rec gcd a b :=
        if b <> 0 then gcd b (a mod b) else return a
```

i) Name of the function – gcd

ii) Identify the statement which tells it is a recursive function – rec keyword

iii) Name of the argument variable – a and b

iv) Statement which invoke the function recursively – gcd b (a mod b)

v)Statement which terminates the recursion – return a

3. Explain with example Pure and impure functions.

Pure functions

* Pure functions are functions which will give exact result when the same arguments are passed.
* The return value of the pure functions solely depends on its arguments passed.
* They do not modify the arguments which are passed to them.

```
let square x:=
        return: x * x
```

* The above function square is a pure function because it will not give different results for same input.

Impure functions

* The return value of the impure functions does not solely depend on its arguments passed.
* They may modify the arguments which are passed to them.

```
let randomnumber :=
        a := random()
        return a
```

* The above function is an impure function because the randon()will give different outputs for the same function call.

4. Explain with an example interface and implementation.

Interface

* Interface just defines what an object can do, but won't actually do it.
* In object oriented programs classes are the interface and how the object is processed and executed is the implementation.
* For example when you press a light switch, the light goes on, you may not have cared how it splashed the light.
* A light, should have function definitions like turn_on () and a turn_off ().

Implementation

* Implementation carries out the instructions defined in the interface.
* For example, the person who drives the car doesn't care about the internal working.

Internally, the engine of the car is doing all the things.

* It's where fuel, air, pressure, and electricity come together to create the power to move the vehicle.
* Thus we separate interface from implementation.

Lesson 2 - Data Abstraction

1. How will you facilitate data abstraction. Explain it with suitable example.

Data abstraction is used to define an Abstract Data Type (ADT), which is a collection of constructors and selectors.

Constructors are functions that build the abstract data type.

city:= makecity (name, lat, lon)

Here makecity (name, lat, lon) is the constructor which creates the object city.

Selectors are nothing but the functions that retrieve information from the data type.

- getname(city)
- getlat(city)
- getlon(city)

are the selectors, because these functions extract the information of the city object.

2. What is a List? Why List can be called as Pairs. Explain with suitable example.

List is constructed by placing expressions within square brackets separated by commas.

Each value can be of any type and can even be another list.

Example for list is lst := [10, 20]

The elements of a list can be accessed in two ways.

I. Multiple assignment

```
lst := [10, 20]
x, y := lst
```

In the above example x will become10 and y will become 20.

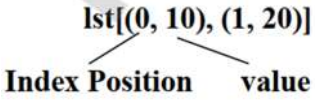
II. Element selection operator

lst[0] is 10 and lst[1] is 20

Any way of bundling two values together into one can be considered as a pair.

Mathematically we can represent list similar to a set.

lst[(0, 10), (1, 20)] – where



Therefore List can be called as Pairs.

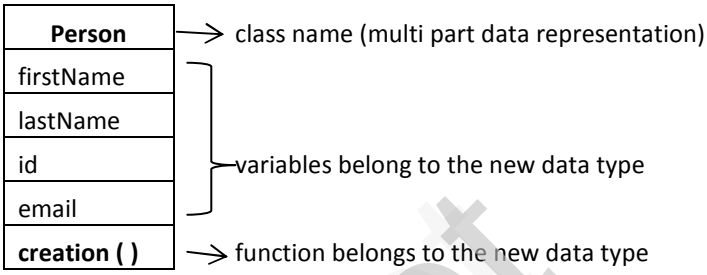
3. How will you access the multi-item. Explain with example.

The structure construct or class construct is used to represent multi-part objects where each part is named.

Consider the following pseudo code:

```
class Person:
        creation( )
        firstName := " "
        lastName := " "
        id := " "
        email := " "
```

The new data type Person is pictorially represented as




```
let main() contains
        p1:=Person()
        firstName := "Rajesh "
        lastName :="Kanna"
        id :="1234"
        email:="compsci@gmail.com"
```

In this example Person is referred to as a class or a type, while p1 is referred to as an object or an instance.

Lesson 3 – Scoping

1 Explain the types of scopes for variable or LEGB rule with example.

Scope refers to the accessibility of a variable with in one part of a program to another part of the same program. There are 4 types of Variable Scope.

Local Scope

Local scope refers to variables defined in current function. Example :-

```
Disp():
        a:=7
        print a
Disp()
```

Output is 7

Global Scope

A variable which is declared outside of all the functions in a program is known as global variable.

Example:-

```
a:=10
Disp():
        a:=7
        print a          → 7
Disp()
print a                  → 10
```

Enclosed Scope

A variable which is declared inside a function which contains another function definition with in it.

Example:-

```
Disp():
        a:=10
        Disp1():
                print a          →10
        Disp1()
        print a                  →10
Disp()
```

Built-in Scope

Any variable or function which is defined in the modules of a programming language has Built-in or module scope.

Example:-

```
Built-in or module scope
Disp():
        a:=10
        print a
Disp()
```

2. Write any Five Characteristics of Modules.

1. Modules contain instructions, logic, and data.

2. Modules can be separately compiled and stored in a library.

3. Modules can be included in a program.

4. Module segments can be used by invoking a name and some parameters.

5. Module segments can be used by other modules

3. Write any five benefits in using modular programming

- Less code to be written.
- Modular programming allows many programmers to collaborate on the same application.
- The code is stored across multiple files.
- Code is short, simple and easy to understand.
- Errors can easily be identified, as they are localized to a subroutine or function.
- The same code can be used in many applications.
- The scoping of variables can easily be controlled

Lesson 4 - Algorithmic strategies

1. Explain the characteristics of an algorithm.

Input – Zero or more quantities to be supplied.

Output – At least one quantity is produced.

Finiteness – Algorithms must terminate after finite number of steps.

Effectiveness – Every instruction must be carried out effectively.

Correctness – The algorithms should be error free.

Simplicity – Easy to implement.

Unambiguous – Algorithm should be clear and unambiguous.

Independent – An algorithm should be independent of any programming code.

2. Discuss about Linear search algorithm.

Linear search is also called sequential search.

This method checks the search element with each element in sequence until the desired element is found.

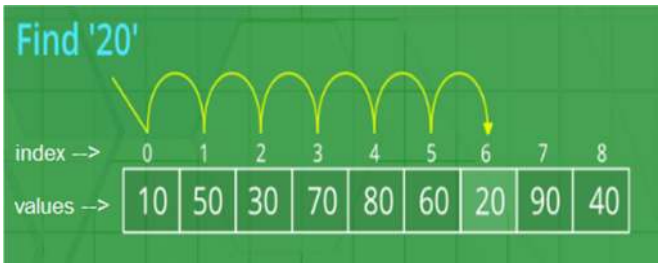
In this searching algorithm, list need not be ordered.

Pseudo code

* Let us assume that, x be the search element,
* Start from the first element of the array and one by one compare x with each element of the array.
* If x matches with an element, return the index.
* If x doesn't match with any of the elements, return -1.

Example:

To search the number 20 in the array given below, linear search will go step bystep in a sequential order starting from the first element.



1) Search Element : 20

Item found at : 6

2) Search Element : 35

Item found at : -1 (not found)

3. What is Binary search? Discuss with example.

Binary search also called half-interval search algorithm.

It finds the position of a search element within a sorted array.

This searching technique follows the divide and conquer approach.

Pseudo code for Binary search

- Start with the middle element.
- If the key is found at middle element, then return in index of middle element.
- If the key is smaller than the middle element, then the left side is used for next search.
- If the key is larger than the middle element, then the right side is used for next search.
- This process is continued until the key is found or no elements to search.

Example:

4. Explain the Bubble sort algorithm with example.

Bubble sort, also known as comparison sort, is the easiest sorting algorithm.

It compares each pair of adjacent elements and swaps them if they are in the unsorted order.

Although the algorithm is simple, it is too slow and less efficient when compared to other sorting methods.

Procedure

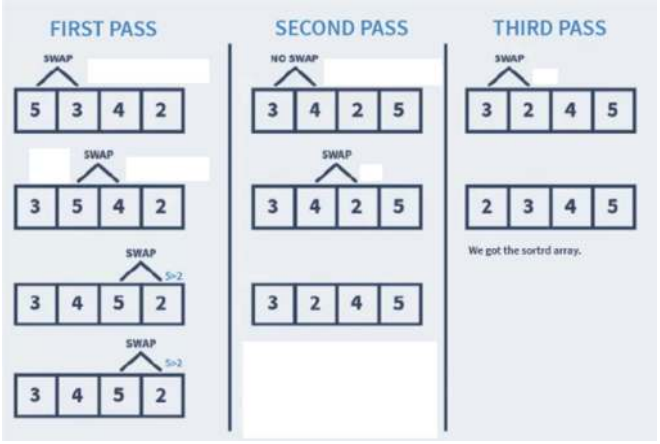
1) Start with the first element.

2) Compare the current element with the next element.

3) If the current element is greater than the next element, then swap both the elements. If not, move to the next element.

4) Repeat steps 1 – 3 until we get the sorted list.

Example:



5. Explain the concept of Dynamic programming with suitable example.

Dynamic programming is a technique that breaks the problems into sub-problems, and saves the result for future purposes so that we do not need to compute the result again.

Steps to do Dynamic programming

- It breaks down the complex problem into simpler sub-problems.
- It finds the optimal solution to these sub-problems.
- It stores the results of sub-problems (memoization).
- Finally, calculate the result of the complex problem.

Dynamic programming approach for Fibonacci series:

Initialize f0=0, f1 =1

step-1: Print the initial values of Fibonacci f0 and f1

step-2: Calculate Fibonacci fib ← f0 + f1

step-3: Assign f0← f1, f1← fib

step-4: Print the next consecutive value of Fibonacci fib

step-5: Go to step-2 and repeat until the specified number of terms generated

For example if we generate Fibonacci series up to 10 digits, the algorithm will generate the series as shown below:

The Fibonacci series is : 0 1 1 2 3 5 8 13 21 34 55

5. PYTHON – VARIABLES AND OPERATORS

1. Describe in detail the procedure of Script mode programming.

Basically, a script is a text file containing the Python statements. It can be executed again and again without retyping.

1. Choose **File → New File** or press **Ctrl + N** in Python shell window.

2. An untitled blank script text editor will be displayed on screen.

3. Type the following code in Script editor

```
a =100
b = 350
c = a+b
print ("The Sum=", c)
```

4. Choose **File → Save** or Press **Ctrl + S**

In the Save As dialog box, type the file name in File Name box

5. Choose **Run → Run Module** or Press **F5** to execute the code.

6. For all error free code, the output will appear in the IDLE window.

2. Explain input() and print() functions with examples.

The print() function

In Python, the print() function is used to display result on the screen. The syntax for print() is as follows:

```
print(message)
print(message, variable)
```

Example:

```
print("Hello Python")
a=5
print("a= ", a)
```

The input() function

In Python, input( ) function is used to accept data as input at run time. The syntax for input() function is,

```
variable = input ("prompt string")
```

Where, prompt string is a message to the user.

Example:

```
name = input("Enter your name: ")
print(name)
```

3. Discuss in detail about Tokens in Python

Tokens

A token is the smallest individual unit in a python program. All statements and instructions in a program are built with tokens.

The normal token types are

1) Keywords,

2) Identifiers,

3) Operators,

4) Delimiters and

5) Literals.

1. Keywords: Keywords are words that have some special meaning. They can't be used for any other purpose.

Ex: False, True, class, break, continue, while, for, import etc.

2. Identifiers: An Identifier is a name used to identify a variable, function, class, module or object. Python is a case-sensitive language.

Example of valid identifiers

sum, total_marks, regno, num1

3. Operators: In computer programming languages operators are special symbols which represent computations, conditional matching etc.

Operators are categorized as Arithmetic, Relational, Logical, Assignment etc.

Value and variables when used with operator are known as operands.

4. Delimiters (Punctuators): These are the symbols that used in Python to organize the structures, statements, and expressions.

Some of the delimiters are: [ ] { } ( ) = : , etc.

5. Literals: Literal is a raw data given to a variable or constant.

In Python, there are various types of literals.

1) Numeric 2) String 3) Boolean

(i) Numeric Literals consist of digits and are immutable (unchangeable).

Ex: rollno=101

pi=3.14

(ii) In Python a string literal is a sequence of characters surrounded by quotes.

Ex: name="Rajesh"

(iii) A Boolean literal can have any of the two values: True or False.

Ex: choice=True

Prepared by

S.Ganesh Kumar , B.Sc.,B.Ed.,M.S.I.T.

N.S.Hr.Sec.School, Theni

CHAPTER 6 CONTROL STRUCTURES

1. Write a detailed note on for loop.

The for loop is usually known as a definite loop. Because the programmer knows exactly how many times the loop will be executed.

Syntax:

```
for variable in sequence :
        statements
```

Here the sequence is the collection of ordered or unordered values or even a string.

The control variable accesses each item of the sequence on each iteration until it reaches the last item.

Example:

```
for x in “Python” :
        print(x)
```

Output

```
P
y
t
h
o
n
```

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

The elif statement enables us to check multiple conditions and execute the specific block of statements.

Syntax:

```
if condition1:
        code block 1
elif condition2:
        code block 2
else:
        code block 3
```

Multiple if..else statements can be combined to one if..elif…else. ‘ elif ’ can be considered to be abbreviation of ‘ else if ’.

Example:

```
number = int(input(“Enter a number”))
if number == 0:
        print("Zero")
elif number > 0:
        print(“Positive number”)
else:
        print('Negative number')
```

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

```
for i in range(101 , 1000 , 2):
        print(i , end = " ")
```

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

```
n=int(input("Enter table number "))
for i in range(1,11):
        print(" %d x %d = %d " %( i , n , i*n ))
```

CHAPTER 7 PYTHON FUNCTIONS

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

Functions are nothing but a group of related statements that perform a specific task.

Function types:

- Built-in Functions
- User-defined Functions
- Lambda Functions
- Recursive Functions

Built-in Functions:

Functions that are inbuilt with in Python.

Ex: max( ) – Returns the maximum value in a list.

User-defined functions

Functions defined by the users themselves.

Function blocks begin with the keyword “def ”

Ex:

```
def hello( ):
        print (“hello - Python”)
        return
```

Lambda functions

Lambda Functions are anonymous functions.

We can use the lambda keyword to define an un-named function.

```
lambda arguments: expression
```

Ex:

```
x = lambda a : a + 10
print(x(5))
```

Recursive Functions

When a function calls itself is known as recursion.

A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

Ex:

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

2. Explain the scope of variables with an example.

Scope of variable refers to the part of the program, where it is accessible.

There are two types of scopes -
local scope and global scope.

Local Scope

A variable declared inside a function belongs to local scope.

A variable with local scope can be accessed only within the function.

Example

```
def myfunc():
        x = 300  ------> local scope
        print(x)
myfunc()
```

Global scope

In Python, a variable declared outside of the function is known as a global variable.

This means that a global variable can be accessed inside or outside of the function.

Example

```
x = 300  ------> global scope
def myfunc( ) :
        print(x)

myfunc( )
```

3. Explain the following built-in functions. (a) id() (b) chr() (c) round() (d) type() (e) pow()

(a) id( )

Returns the "identity" or the address of the object in memory.

Ex: x=15
print ("address of x is :" , id (x))
Output: address of x is : 1357486752

(b) chr( )

Returns the Unicode character for the given ASCII value.

Ex: c = 65
print (chr (c))
Output: A

(c) round( )

round() function that rounds off a number to the given number of digits.

Ex: average=76.32
Print(round(average,1))
Output: 76.3

(d) type( )

Returns the type of data stored in the objects or variables.

Ex: x= 15.2
print (type (x))
Output: <class 'float'>

(e) pow ()

This function is used to compute the powers of a number. It returns x to the power of y.

Ex: x=5
y=2
Print(pow(x,y))
Output: 25

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

```
num1 = int(input(" Enter First number : "))
num2 = int(input(" Please Second number : "))
a = num1
b = num2

while(num2 != 0):
        temp = num2
        num2 = num1 % num2
        num1 = temp

gcd = num1
lcm = (a * b) / gcd

print("\n GCD = ",gcd)
print("\n LCM = ",lcm)
```

5. Explain recursive function with an example.

When a function calls itself is known as recursion.

A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

Ex:

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


print(fact(6))
```

The recursion ends when the number reduces to 1. This is called the base condition.

CHAPTER 8 STRINGS AND STRING MANIPULATION

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

Python provides the following operators for string operations. These operators are useful to manipulate string.

(i) Concatenation (+)

Joining of two or more strings is called as Concatenation.

The plus (+) operator is used to concatenate strings in python.

Example

```
str1= "welcome" + "Python"
print(str1)
Output:   'welcomePython'
```

(ii) Append (+ =)

Adding more strings at the end of an existing string is known as append.

The operator += is used to append a new string with an existing string.

Example

```
str1="Hello "
str1 += " Python"
print (str1)
Output:   Hello 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 Welcom
```

(iv) String slicing

Slice is a substring of a main string. A substring can be taken from the original string by using [ ] operator.

General format of slice operation: str[start:end]

Example

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

CHAPTER 9 LISTS, TUPLES, SETS AND DICTIONARY

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

There are three methods to add elements to a List in Python.

Append( )

The append() method is used to add an element to the end of the list.

```
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)
Output: [1, 2, 3, 4]
```

extend( )

extend() function is used to add more than one element to an existing list.

```
my_list = [1, 2, 3]
my_list.extend( [4 , 5] )
print(my_list)
Output: [1, 2, 3, 4, 5]
```

insert( )

The insert() function is used to insert an element at any position of a list.

```
my_list = ["apple", "banana", "cherry"]
my_list.insert(1,"mango")
print(my_list)
Output: ['apple', 'mango', 'banana', 'cherry']
```

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

The range() function is used to generate a series of values in Python.

The range() function has three arguments.

```
range (start , stop, step)
```

where,

start - Optional. Start value of the series. Default is 0

stop - Required. End value of the series (not included)

step - Optional. Increment value of the series. Default is 1

Example : Generating first 10 even numbers

```
for x in range (2, 11, 2):
    print(x)
```

Output

```
2
4
6
8
10
```

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

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

Output:

('Raja', 'XII-F', 98.7)

('Mani', '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.

As you learnt in mathematics, the python is also supports the set operations such as Union, Intersection, difference and Symmetric difference.

(i) Union:

It includes all elements from two or more sets.

In python, the operator | and the function union() are used to join two sets in python.

```
A={1,2,3,4}
B={3,4,5,6}
print(A|B)  or  print(A.union(B))
```

output: {1, 2, 3, 4, 5, 6}

(ii) Intersection:

It includes the common elements in two sets.

In python, the operator & and the function intersection() are used to intersect sets in python.

```
Ex:  A={1,2,3,4}
     B={3,4,5,6}
     print(A&B)  or  print(A.intersection(B))
```

output: {3, 4}

(iii) Difference

It includes all elements that are in first set but not in the second set.

The minus (-) operator and the function difference() are used to do difference operation.

```
Ex:  A={1,2,3,4}
     B={3,4,5,6}
     print(A – B)  or  print(A.difference(B))
```

output: {1, 2}

(iv) Symmetric difference

The symmetric difference between two sets includes all elements without the common elements.

In Python, we use the caret ^ operator or the symmetric_difference() method to perform symmetric difference between two sets.

```
Ex: A={1,2,3,4}
    B={3,4,5,6}
    print(A^B)  or  print(A.symmetric_difference(B))
```

output: {1, 2, 5, 6}

CHAPTER 10 PYTHON CLASSES AND OBJECTS

1. Explain about constructor and destructor with suitable example.

Constructor

A constructor is a special type of function which is used to initialize the members of the class.

In Python the __init__() method is called the constructor and is always called when an object is created.

Destructor

Destructor is also a special method to destroy the objects.

In Python, __del__( ) method is used as destructor. It is just opposite to constructor.

Example

```
class sample( ) :
        def __init__(self) :
                print("it a constructor")
        def __del__(self) :
                print("it a destructor")
obj = sample()
```

CHAPTER 11 DATABASE CONCEPTS

1. Explain the different types of data model.

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

Hierarchical Model

Hierarchical model was developed by IBM as Information Management System.

In Hierarchical model, data is represented as a simple tree like structure form.

This model represents a one-to-many relationship i.e parent-child relationship.

Network Model

Network database model is an extended form of hierarchical data model. The difference between hierarchical and Network data model is :

• In hierarchical model, a child record has only one parent node.

• In a Network model, a child may have many parent nodes. It represents the data in manyto-many relationships.

• This model is easier and faster to access the data.

Relational Model

This model was introduced by **E.F Codd** in 1970.

The basic structure of data in the relational model is **tables**.

The relationship is maintained by storing a common field.

Hence, tables are also known as relations.

Entity Relationship Model. (ER model)

It was developed by Chen in 1976.

Relationship are created by dividing the object into entity and characteristics into attributes.

It is very simple and easy to design logical view of data.

The developer can easily understand the system by looking at ER model.

Object model

Object model stores the data in the form of objects, attributes and methods.

This model handles more complex applications, such as Geographic information System.

It is used in file Management System.

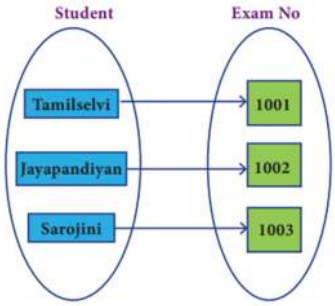
It represents real world objects, attributes and behaviors.

2. Explain the different types of relationship mapping.

One-to-One Relationship

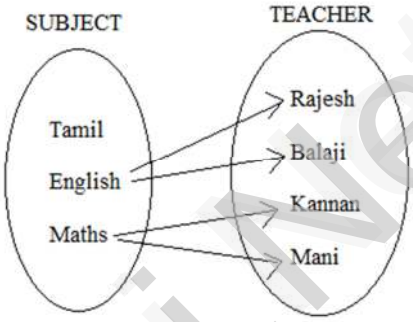
In One-to-One Relationship, one entity is related with only one other entity.

For example: A student can have only one exam number.



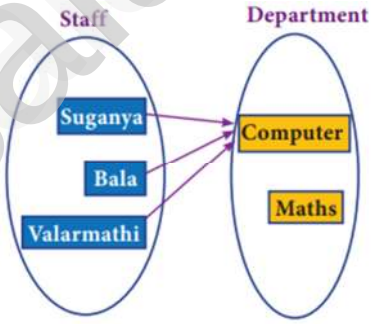
One-to-Many Relationship

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.



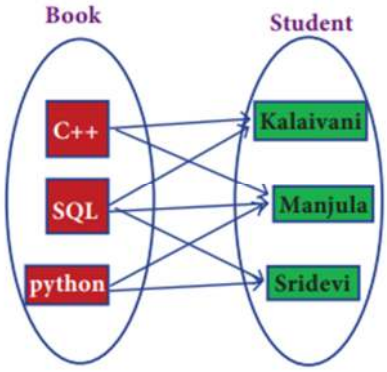
Many-to-One Relationship

In Many-to-One Relationship, many entities can be related with only one in the other entity.



Many-to-Many Relationship

A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table.



3. Differentiate DBMS and RDBMS.

Basis of Comparison	DBMS	RDBMS
Expansion	Database Management System	Relational Database Management System
Data storage	Navigational model	Relational model (in tables).
Data redundancy	Present	Not Present
Normalization	Not performed	RDBMS uses normalization to reduce redundancy
Data access	Consumes more time	Faster, compared to DBMS.
Transaction management	Inefficient and insecure.	Efficient and secure
Distributed Databases	Not supported	Supported by RDBMS.
Example	Dbase, FoxPro.	Oracle, MySql, SQLite

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

Relational Algebra is a procedural query language used to query the database tables using SQL.

SELECT (symbol : σ)

The select operation selects tuples that satisfy a given condition.

Example:

σ_{course} = "Big Data" (STUDENT)

PROJECT (symbol : Π)

This operation shows the list of attributes that we wish to appear in the result.

Rest of the attributes are eliminated from the table.

Example:

∏ RollNo, Name (STUDENT)

UNION (Symbol :∪)

It includes all tuples that are in tables A or in B.

It also eliminates duplicates.

Example:

∏ RollNo, Name (STUDENT) U ∏ Rank (MARKS)

INTERSECTION (symbol : ∩)

Defines a relation consisting of a set of all tuple that are in both in A and B.

Example:

∏ RollNo, Name (STUDENT) ∩ ∏ RollNo (MARKS)

SET DIFFERENCE (Symbol –)

The result of A – B, is a relation which includes all tuples that are in A but not in B.

Example:

∏ RollNo (STUDENT) – ∏ RollNo (MARKS)

PRODUCT OR CARTESIAN PRODUCT (Symbol : X)

Cross product is a way of combining two relations. The resulting relation contains, both relations being combined.

Example:

(STUDENT) X (MARKS)

5. Explain the characteristics of RDBMS.

	RDBMS
Expansion	Relational Database Management System
Data storage	Relational model (in tables).
Data redundancy	Not Present
Normalization	RDBMS uses normalization to reduce redundancy
Data access	Faster, compared to DBMS.
Transaction management	Efficient and secure
Distributed Databases	Supported by RDBMS.
Example	Oracle, MySql, SQLite

CHAPTER 12 - STRUCTURED QUERY LANGUAGE (SQL)

1. Write the different types of constraints and their functions.

SQL Constraints

Constraint is a condition applicable on a field or set of fields.

Column constraint: Column constraint apply only to individual column.

Table constraint : Table constraint apply to a group of one or more columns.

Type of Constraints:

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

CHECK - Ensures that the values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column if no value is specified

Example:

CREATE TABLE Student (Adno integer PRIMARY KEY

Name char(20) NOT NULL,

Age integer CHECK (Age <=19),

Place char(10) DEFAULT "Theni"

);

2. Consider the following employee table. Write SQL commands for the questions.(i) to (v).

EMPCODE	NAME	DESIG	PAY	ALLOWANCE
S1001	Hariharan	Supervisor	29000	12000
P1002	Shaji	Operator	10000	5500
P1003	Prasad	Operator	12000	6500
C1004	Manjima	Clerk	8000	4500
M1005	Ratheesh	Mechanic	20000	7000

(i) To display the details of all employees in descending order of pay.

SELECT *FROM employee ORDER BY PAY DESC;

(ii) To display all employees whose allowance is between 5000 and 7000.

SELECT *FROM employee WHERE ALLOWANCE BETWEEN 5000 AND 7000;

(iii) To remove the employees who are mechanic.

DELETE FROM employee WHERE DESIG=”Mechanic”;

(iv) To add a new row.

INSERT INTO employee VALUES (“M1006”,”Rajesh”,”Mechanic”,20000,7000);

(v) To display the details of all employees who are operators.

SELECT *FROM employee WHERE DESIG=”Operator”;

3. What are the components of SQL? Write the commands in each.

SQL commands are divided into five categories:

1) DDL - Data Definition Language

It consist of SQL statements used to define the database structure or schema.

CREATE - To create tables in the database.

ALTER - Alters the structure of the database.

DROP - Deletes tables from database.

2) DML - Data Manipulation Language

DML is a query language used for adding , deleting, and updating data in a database.

INSERT - Inserts data into a table.

UPDATE - Updates the existing data within a table.

DELETE - Deletes all records from a table, but not the space occupied by them.

DCL - Data Control Language

DCL is a language used to control the access of data stored in a database.

GRANT - It is used to give user access privileges to a database.

REVOKE - It is used to take back permissions from the user.

TCL - Transaction Control Language

TCL commands are used to manage transactions in the database.

COMMIT - Saves any transaction into the database permanently.

ROLL BACK - Restores the database to last commit state.

DQL - Data Query Language

It consist of commands used to query or retrieve data from a database.

SELECT - Displays the records from the table.

4. Construct the following SQL statements in the student table-

(i) SELECT statement using GROUP BY clause.

The GROUP BY statement groups rows that have the same values.

The GROUP BY statement is often used with aggregate functions.

For example- To count the number of male and female students in the student table, the following command is given :

SELECT Gender, count(*) FROM Student GROUP BY Gender;

(ii) SELECT statement using ORDER BY clause.

The ORDER BY clause in SQL is used to sort the data in either ascending or descending based on one or more columns.

For example: To display the students in alphabetical order of their names, the command is used as

SELECT * FROM Student ORDER BY Name;

5. Write a SQL statement to create a table for employee having any five fields and create a table constraint for the employee table

```
CREATE TABLE employee (
empcode INTEGER NOT NULL,
empname CHAR(20),
desig CHAR(15),
pay INTEGER,
allowance INTEGER,
PRIMAY KEY(empcode)
);
```

CHAPTER 13 - PYTHON AND CSV FILES

1. Differentiate Excel file and CSV file.

Excel	CSV
MS Excel stands for Microsoft Excel.	CSV stands for Comma separated value.
Excel is a binary file that holds information, including both content and formatting.	It is a plain text format with a series of values separated by commas.
Excel file can be opened with Microsoft Excel only.	CSV can be opened with any text editor.
Excel file saved with extension as .xls / .xlsx	CSV file saved with extension as .csv
It consumes more memory.	It consumes less memory.

2. Tabulate the different mode with its meaning.

Python File Modes

Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing.
'x'	Open a file for exclusive creation.
a'	Open for appending at the end of the file.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

3. Write the different methods to read a File in Python.

There are two ways to read a CSV file.

1. Using the csv module's reader function

2. Using the DictReader class.

CSV Module's Reader Function

We can read the contents of CSV file with the help of csv.reader() function. The syntax is,

csv.reader(fileobject, delimiter, parameters)

Ex.

```
import csv
F=open("c:\pyprg\student.csv" , 'r')
reader = csv.reader(F)
for row in reader:
        print(row)
F.close()
```

Using the DictReader class

The csv.DictReader() method can be used to read a CSV file as a dictionary.

Ex:

```
import csv
F=open("c:\pyprg\student.csv" , 'r')
reader = csv.DictReader(F)
for row in reader:
        print( dict(row) )
F.close()
```

4. Write a Python program to write a CSV File with custom quotes.

```
import csv
csv.register_dialect( 'myDialect' , quoting=csv.QUOTE_ALL)
F= open("c:\pyprg\student.csv",  "w")
writer = csv.writer(F, dialect="myDialect")
writer.writerow( ['Adno', 'Name, 'Class'] )
writer.writerow([121, 'Rajesh', 'XII-IC'] )
F.close()
```

5. Write the rules to be followed to format the data in a CSV file.

(i) Each record (row of data) is to be located on a separate line.

(ii) The last record in the file may or may not have an ending line break.

(iii) There may be an optional header line appearing as the first line of the file.

(iv) Within the header and each record, there may be one or more fields, separated by commas.

(v) Each field may or may not be enclosed in double quotes.

(vi) Fields containing line breaks and commas should be enclosed in double-quotes.

CHAPTER 14 - IMPORTING C++ PROGRAMS IN PYTHON

1. Write any 5 features of Python.

- Python uses Automatic Garbage Collection whereas C++ does not.
- C++ is a statically typed language, while Python is a dynamically typed language.
- Python runs through an interpreter, while C++ is pre-compiled.
- Python code tends to be 5 to 10 times shorter than that written in C++.
- In Python, there is no need to declare types explicitly where as it should be done in C++
- In Python, a function may return multiple values. But in C++, a function can return only one value.

2. Explain each word of the following command.

python <filename.py> -<i> <C++ filename without cpp extension>

python	keyword to execute the Python program from commandline
filename.py	Name of the Python program to executed
- i	input mode
C++ filename without cpp extension	name of C++ file to be compiled and executed

3. What is the purpose of sys, os, getopt modules in Python. Explain

sys module

This module provides access to built-in variables used by the interpreter.

One among the variable in sys module is argv

sys.argv is the list of command-line arguments passed to the Python program.

To use sys.argv, import sys should be used. The first argument, sys.argv[0] contains the name of the python program.

OS Module

The OS module in Python provides functions for interacting with the operating system.

The os.system() method executes the command in a subshell.

os.system(command)

Example to find the current date of computer:

```
import os
print (os.system('date'))
```

The getopt Module

The getopt module contains functions to extract command-line options and arguments.

This module provides getopt() method to enable command-line argument parsing.

It can handle both short and long option formats.

Syntax:

getopt.getopt (args , options , [long_options])

4. Write the syntax for getopt() and explain its arguments and return values

The getopt module of Python helps you to parse (split) command-line options and arguments.

This module provides getopt() method to enable command-line argument parsing.

Syntax:

getopt.getopt (args , options , [long_options])

args: The args is the list of arguments which are to be passed

options: The options are the string of options letters which should be written with a colon (:).

long_options: This is the list of the string which have the name of long options should be written with an equal sign (=).

Return Type: The first element of the return value is the list of (option , value) pairs, and the second element of the return value is a list of program arguments which are left.

Ex: opts, args = getopt.getopt (argv, "i:",['ifile='])

5. Write a Python program to execute the following c++ coding

```
#include <iostream>
using namespace std;
int main()
{ cout<<“WELCOME”;
return(0);
}
```

The above C++ program is saved in a file welcome.cpp

```
import os
argv = sys.argv[1:]
cppfile = ''
exefile = ''

opts, args = getopt.getopt(argv, "i:o:", ["ifile="])
```

```
for opt, arg in opts:
   if opt == '-i':
      cppfile = arg+".cpp"
   elif opt == '-o':
      exefile = arg+".exe"

os.system('g++ ' + cppfile + ' -o ' + exefile)
os.system(exefile)
```

CHAPTER 15 DATA MANIPULATION THROUGH SQL

1. Write in brief about SQLite and the steps used to use it.

- ✓ SQLite is a simple relational database system.
- ✓ It saves data in regular data files within internal memory of the computer.
- ✓ It is designed to be embedded in applications.
- ✓ SQLite is fast and flexible, making it easier to work.
- ✓ Python has a native library for SQLite.

To use SQLite,

Step 1 : import sqlite3

Step 2 : create a connection using connect () method.

Step 3 : Set the cursor object cursor = connection. cursor ()

2. Write the Python script to display all the records of the following table using fetchmany()

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000
1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

Python Script:

```
import sqlite3
connection = sqlite3.connect ("shop.db")
cursor = connection . cursor ()
cusrsor. execute ("SELECT * FROM stock")
result = cursor. fetchmany (5)
print(*result,sep="\n")
```

3. What is the use of HAVING clause. Give an example python script.

Having clause is used to filter data based on the group functions.

This is similar to WHERE condition but can be used only with group functions.

```
import sqlite3
connection = sqlite3.connect("school.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall( )
print(result)
```

4. Write a Python script to create a table called ITEM with following specification. Add one record to the table.

Name of the database :- ABC
Name of the table :- Item
Column name and specification :-
Icode :- integer and act as primary key
Item Name :- Character with length 25
Rate :- Integer
Record to be added :- 1008, Monitor,15000

```
import sqlite3
connection = sqlite3 . connect ("ABC.db")
cursor = connection . cursor ()
qry = " CREATE TABLE Item (Icode INTEGER, Item_Name VARCHAR (25), Rate Integer);"
cursor.execute (qry)
qry = "INSERT INTO Item VALUES (1008, 'Monitor', 15000);"
cursor.execute (qry)
connection.commit ()
connection.close ()
```

5. Consider the following table Supplier and item .Write a python script for (i) to (ii)

Suppno	Name	City	Icode	SuppQty
S001	Prasad	Delhi	1008	100
S002	Anu	Bangalore	1010	200
S003	Shahid	Bangalore	1008	175
S004	Akila	Hydrabad	1005	195
S005	Girish	Hydrabad	1003	25
S006	Shylaja	Chennai	1008	180
S007	Lavanya	Mumbai	1005	325

i) Display Name, City and Item name of suppliers who do not reside in Delhi.

ii) Increment the SuppQty of Akila by 40

(i) import sqlite3

connection = sqlite3.connection ("supplier.db")

cursor = connection.cursor ()

cursor.execute ("SELECT Name, City, Icode FROM Item WHERE City <> "Delhi")

result = cursor.fetchall ()

print (* result, sep = "\n")

(ii) import sqlite3

conn = sqlite3.connect ("supplier.db")

conn.execute ("UPDATE Item SET Name = 'Akila' WHERE SuppQty = 235)

conn.commit ()

cursor = conn.execute ("SELECT * FROM Item")

for row in cursor:

print (row)

conn.close ()

CHAPTER 16 DATA VISUALIZATION USING PYPLOT

1. Explain in detail the types of pyplots using Matplotlib.

Line Chart

A Line Chart or Line Graph is a type of chart which displays information as a series of data points connected by straight line segments.

Example:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()
```

Bar Chart

A BarChart is one of the most common type of plot.

It shows the relationship between a numerical data and a categorical values.

Bar chart represents data with rectangular bars.

Example:

```
import matplotlib.pyplot as plt
subjects = ["TAMIL", "ENGLISH", "MATHS"]
marks = [80, 60, 85]
plt.bar (subjects, marks)
plt.show()
```

Pie Chart

Pie Chart is probably one of the most common type of chart.

It is a circular graphic which is divided into slices to illustrate numerical proportion.

Example

```
import matplotlib.pyplot as plt
subjects = ["TAMIL", "ENGLISH", "MATHS"]
marks = [80, 60, 85]
plt.pie (marks,labels=subjects, autopct = "%.2f")
plt.show()
```

2. Explain the various buttons in a matplotlib window.

(i) Home Button → The Home Button will help one to begun navigating the chart. If you ever want to return back to the original view, you can click on this.

(ii) Forward/Back buttons → These buttons can be used like the Forward and Back buttons in browser. Click these to move back to the previous point you were at, or forward again.

(iii) Pan Axis → This cross-looking button allows you to click it, and then click and drag graph around.

(iv) Zoom → The Zoom button lets you click on it, then click and drag a square would like to zoom into specifically. Zooming in will require a left click and drag. Zoom out with a right click and drag.

(v) Configure Subplots → This button allows you to configure various spacing options with figure and plot.

3. Explain the purpose of the following functions: a. plt.xlabel b. plt.ylabel c. plt.title d. plt.legend() e. plt.show()

a. plt.xlabel - Specifies label for x-axis

b. plt.ylabel - Specifies label for y-axis

c. plt.title - Specifies title to the graph.

d. plt.legend() - Places a legend on various types of graphs.

e. plt.show() - Displays the plot.

----- & -----

Prepared by:

S.Ganesh Kumar, B.Sc.,B.Ed.,M.S.I.T.,

N.S.Hr.Sec.School, Theni.

(for any suggestions please mail to rsgk05@gmail.com)

Also available at www.scribd.com/rsgk