SHRI VIDHYABHARATHI MATRIC HR.SEC.SCHOOL

SAKKARAMPALAYAM , AGARAM (PO) ELACHIPALAYAM

TIRUCHENGODE(TK), NAMAKKAL (DT) PIN-637202

Cell :99655-31727, 99655-35967,94422-88402, 80726-68664

PUBLIC EXAMINATION-2024

STD: XII	TENTATIVE ANSWER KEY	DATE :08.03.24
SUBJECT: COMPUTER SCIENCE		MARKS : 70

| Q. NO	CONTENT	MARK
	PART-I	**15X1=15**
	CHOOSE THE CORRECT ANSWER:	
1	C) Big O	1
2	c) Abstract datatype	1
3	b) F5	1
4	a) Interface	1
5	a) Access control	1
6	d) []	1
7	c) Instantiation	1
8	d) x%4==0	1
9	b) σ	1
10	a) [10,20,35,40,50]	1
11	b)Flat File	1
12	d)select	1
13	a)0 1 3 4	1
14	d)Dash board	1
15	d)ORDER BY	1
	PART - II	**6X2=12**
16	❖ Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of value and a set of operations. ❖ The process of providing only the essentials and hiding the details is known as abstraction. ❖ Abstraction provides modularity (modularity means splitting a program in to many modules). Classes (structures) are the representation for "Abstract Data Types", (ADT).	**2**

17	❖ Arithmetic operators ❖ Relational or Comparative operators ❖ Logical operators ❖ Assignment operators ❖ Conditional operator	2
18	❖ Searching is designed to check for an element or retrieve an element from any data structure where it is stored. **Types:** ❖ Linear (or) sequential search ❖ Binary (or) half interval search	2
19	❖ User-defined functions ❖ Built-in functions ❖ Lambda functions ❖ Recursion functions	2
20	❖ Line plot ❖ Scatter plot ❖ Histogram ❖ Box plot ❖ Bar chart ❖ Pie chart	2

21	Hierarchical data model	Network data model	2
	A child record has only one parent node.	A child may have many parent nodes.	
	This model represents a one-to-many relationship ie parent-child relationship.	It represents the data in many-to-many relationships.	
	This model is mainly used in IBM Main Frame computers.	This model is easier and faster to access the data.	

22	❖ A CSV fi le is a human readable text file where each line has a number of fields, separated by commas or some other delimiter**.**	2
23	result = cursor.fetchall() ❖ Example:The fetchall() method is used to fetch all rows from the database table	2
24	❖ pop( ) function can also be used to delete an element using the given index value. pop( ) function deletes and returns the last element of a list if the index is not given.	2

	PART – III		6X3=18
25	**Pure Function**	**Impure Function**	**3**
	The return value of the pure functions solely depends on its arguments passed. Hence, if you call the pure functions with the same set of arguments, you will always get the same return values. They do not have any side effects.	The return value of the impure functions does not solely depend on its arguments passed. Hence, if you call the impure functions with the same set of arguments, you might get the different return values For example, random(), Date().	
	They do not modify the arguments which are passed to them	They may modify the arguments which are passed to them	

26	❖ The elements of a list can be accessed in two ways. The first way is via our familiar method of multiple assignment, which unpacks a list into its elements and binds each element to a different name.
	❖ **Example:**
	lst := [10, 20]
	x, y := lst
	❖ A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.
	❖ **Example:**
	    lst[0]
	    10
	    lst[1]
	    20
			3

27	❖ Asymptotic Notations are languages that uses meaningful statements about time and space complexity. ❖ The following three asymptotic notations are mostly used to represent time complexity of algorithms: **(i)Big O:** ❖ Big O is often used to describe the worst-case of an algorithm. **(ii)Big Ω:** ❖ Big Omega is the reverse Big O, if Bi O is used to describe the upper bound (worst - case) of a asymptotic function, Big Omega is used to describe the lower bound (best-case). **(iii)Big Θ :** ❖ When an algorithm has a complexity with lower bound = upper bound, say that an algorithm has a complexity O (n log n) and Ω (n log n), it's actually has the complexity Θ (n log n), which means the running time of that algorithm always falls in n log n in the best-case and worst-case.		**3**

28	a= int (input("Enter number 1")
	            b= int (input ("Enter number 2")
	            c= int (input ("Enter number 3")
	                    if a>b and a>c:
	                            put ("A is greater")
	                    elif b>a and b>c:
	                    print ("B is greatest")
	                    else:
	                            print ("C is greater")
			3

29	**Description:**Used to capitalize the first character of the string.
	Example:
	>>> city="chennai"
	>>> print(city.capitalize())
	Chennai
	(b) swapcase( )
	Description: It will change case of every character to its opposite case vice-versa.
	Example:
	>>> str1="tAmiL NaDu"
	>>> print(str1.swapcase())
	TaMIl nAdU
		3

30	❖ Constructor is the special function that is automatically executed when an object of a class is created. In Python, there is a special function called **"init"** which act as a Constructor.
	
	❖ It must begin and end with double underscore. This function will act as an ordinary function; but only difference is, it is executed automatically when the object is created.
	
	❖ This constructor function can be defined with or without arguments. This method is used to initialize the class variables.
	Syntax:
	
	    General format of __init__ method (Constructor function)
	
	    def __init__(self, [args ....]):
	
	    <statements>
	
	Example:
	    class Sample:
	        def __init__(self, num):
	            print("Constructor of class Sample...")
	            self.num=num
	            print("The value is :", num)
	    S=Sample(10)
	
	DESTRUCTOR:
	• Destructor is also a special method gets executed automatically when an object exit from the scope. It is just opposite to constructor. In Python, __del__( ) method is used as destructor.
	    class Sample:
	        num=0
	        def __init__(self, var):
	            Sample.num+=1
	            self.var=var
	            print("The object value is = ", var)
	            print("The value of class variable is= ", Sample.num)
	        def __del__(self):
	            Sample.num-=1
	            print("Object with value %d is exit from the scope"%self.var)
	    S1=Sample(15)
	    S2=Sample(35)
	    S3=Sample(45)
		3

31	❖ To automate certain tasks in a program ❖ Extracting information from a data set ❖ Less code intensive as compared to traditional programming language ❖ can bring new functions to applications and glue complex systems together.	3
32	❖ The WHERE clause is used to extract only those records that fulfill a specified condition. In this example we are going to display the different grades scored by male students from "student table" **Python statement:** cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")	3
33	**(i)COMMIT command :** ❖ The **COMMIT** command is used to permanently save any transaction to the database. **(ii) ROLLBACK command** ❖The **ROLLBACK** command restores the database to the last commited state. **(iii)SAVEPOINT command** ❖ The **SAVEPOINT** command is used to temporarily save a transaction so that you can rollback to the point whenever required.	3
	PART - IV	**5X5=25**

34	❖ To facilitate data abstraction, you will need to create two types of functions: constructors and selectors.
	Constructors and Selectors:
	• Constructors are functions that build the abstract data type.
	• Selectors are functions that retrieve information from the data type.
	   For example, say you have an abstract data type called city. This city object will hold the city's name, and its latitude and longitude. To create a city object, you'd use a function like **city = makecity (name, lat, lon)**
	   To extract the information of a city object, you would use functions like
	getname(city)
	getlat(city)
	getlon(city)
	city = makecity (name, lat, lon)
	
	Here make city (name, lat, lon) is the constructor which creates the object city.
	• Selectors are nothing but the functions that retrieve information from the data type. Therefore in the above code
	getname(city)
	getlat(city)
	getlon(city)
	are the selectors because these functions extract the information of the city object.
		5
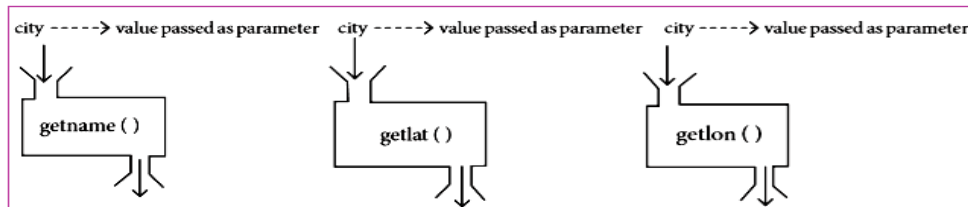
(OR)	❖ Binary search also called half-interval search algorithm. It finds the position of a search element within a sorted array.	

❖ Binary search also called half-interval search algorithm. It finds the position of a search element within a sorted array.

❖ The binary search algorithm can be done as divide-and-conquer search algorithm and executes in logarithmic time.

Pseudo code for Binary search

1. Start with the middle element:

  ❖ If the search element is equal to the middle element of the array i.e., the middle value = number of elements in array/2, then return the index of the middle element.

  ❖ If not, then compare the middle element with the search value,

  ❖ If the search element is greater than the number in the middle index, then select the elements to the right side of the middle index, and go to Step-1.

  ❖ If the search element is less than the number in the middle index, then select the elements to the left side of the middle index, and start with Step1.

2. When a match is found, display success message with the index of the element matched.

  3. If no match is found for all comparisons, then display unsuccessful message.

Binary Search Working principles

❖ List of elements in an array must be sorted first for Binary search. The following example describes the step by step operation of binary search.

❖ Consider the following array of elements, the array is being sorted so it enables to do the binary search algorithm. Let us assume that the search element is 60 and we need to search the location or index of search element 60 using binary search.
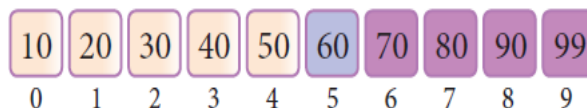

❖ First, we find index of middle element of the array by using this formula :

$$mid = low + (high - low) / 2$$

❖ Here it is, $0 + (9 - 0) / 2 = 4$ (fractional part ignored). So, 4 is the mid value of the array.
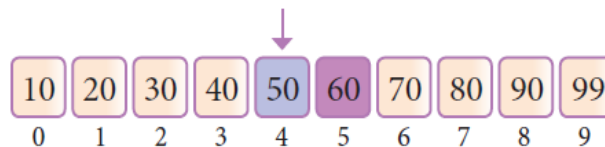
❖ Now compare the search element with the value stored at mid value location 4. The value stored at location or index 4 is 50, which is not match with search element. As the search value 60 is greater than 50.
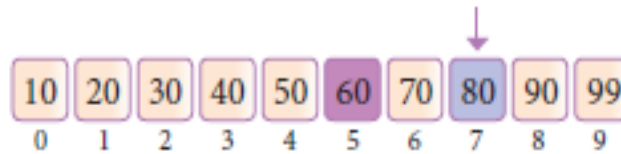

❖ Now we change our low to mid + 1 and find the new mid value again using the

5

formula.

low to mid + 1

mid = low + (high - low) / 2

❖ Our new mid is 7 now. We compare the value stored at location 7 with our target value 31.

❖ The value stored at location or index 7 is not a match with search element, rather it is more than what we are looking for. So, the search element must be in the lower part from the current mid value location
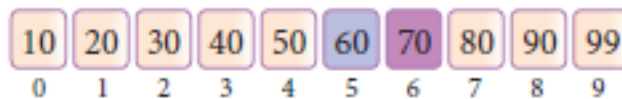
❖ The search element still not found. Hence, we calculated the mid again by using the formula.

high = mid -1

mid = low + (high - low)/2

Now the mid value is 5.

❖ Now we compare the value stored at location 5 with our search element. We found that it is a match.

We can conclude that the search element 60 is found at location or index 5. For example if we take the search element as 95, For this value this binary search algorithm return unsuccessful result.

35	❖ A program needs to interact with the user to accomplish the desired task; this can be achieved using **Input-Output functions**.	
	❖ The **input()** function helps to enter data at run time by the user and the output function **print()** is used to display the result of the program on the screen after execution.	
	Input() function:	
	❖ In Python, **input( )** function is used to accept data as input at run time.	**5**
	❖ The syntax for **input()** function is, **Variable = input ("prompt string")**Where, **prompt string** in the syntax is a statement or message to the user, to know what input can be given. If a prompt string is used, it is displayed on the monitor; the user can provide expected data from the input device.	

	❖ The **input( )** takes whatever is typed from the keyboard and stores the entered data in the given variable. If prompt string is not given in **input( )** no message is displayed on the screen, thus, the user will not know what is to be typed as input.
	Example 1: input ( ) with prompt string
	        >>> city=input ("Enter Your City: ")
	        Enter Your City: Madurai
	        >>> print ("I am from ", city)
	        I am from Madurai
	❖ Note that in example-2, the **input( )** is not having any prompt string, thus the user will not know what is to be typed as input. If the user inputs irrelevant data as given in the above example, then the output will be unexpected. So, to make your program more interactive, provide prompt string with **input( ).**
	❖ The **input ( )** accepts all data as string or characters but not as numbers. If a numerical value is entered, the input values should be explicitly converted into numeric data type. The **int( )** function is used to convert string data as integer data explicitly. We will learn about more such functions in later chapters.
	Example:
	        x = int (input("Enter Number 1: "))
	        y = int (input("Enter Number 2: "))
	        print ("The sum = ", x+y)
	Output:
	        Enter Number 1: 34
	        Enter Number 2: 56
	        The sum = 90
	The print() function
	❖ In Python, the **print()** function is used to display result on the screen. The syntax for **print()** is as follows:
	Example: print ("string to be displayed as output ")
	              print (variable)
	              print ("String to be displayed as output ", variable)
	              print ("String1 ", variable, "String 2", variable, "String 3" ......)
	❖ The **print ( )** evaluates the expression before printing it on the monitor. The print () displays an entire statement which is specified within print ( ).
	❖ **Comma (,)** is used as a separator in **print ( )** to print more than one item.	

(b)	❖ 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. ❖ 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.	**5**

❖ A local variable only exists while the function is executing.

❖ The formate arguments are also local to function.

Example : Create a Local Variable

```
def loc():
        y=0
        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 :

Example :

```
c = 1
def add():
        print(c)
        add()
```

| 36 a) | ❖ 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)
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)
    [2, 4, 6, 8, 10] | 5 |

36	❖ The python is also supports the set operations such as Union, Intersection, difference and Symmetric difference.	
b)	**(i) Union:** It includes all elements from two or more sets	
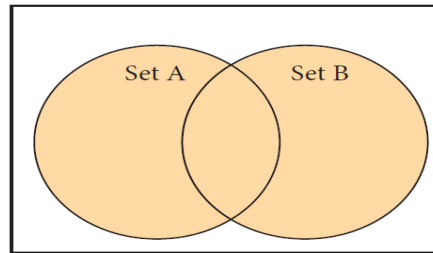
❖ In python, the operator | is used to union of two sets. The function union( ) is also used to join two sets in python.
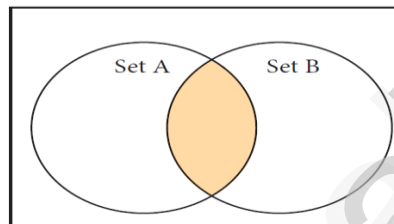
Example: Program to Join (Union) two sets using union operator
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)

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

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

5

❖ The operator & is used to intersect two sets in python. Th e 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 fi rst set (say set A) but not in the second set (say set B)
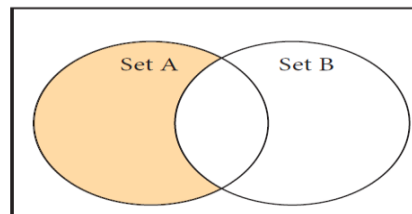
❖ The minus (-) operator is used to difference set operation in python. The function **difference( )** is also used to difference operation.

Example: Program to difference of two sets using minus operator
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.
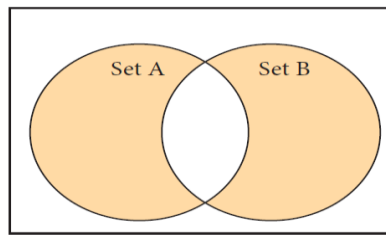
❖ The caret (^) operator is used to symmetric difference set operation in python. The
function **symmetric_difference( )** is also used to do the same operation.
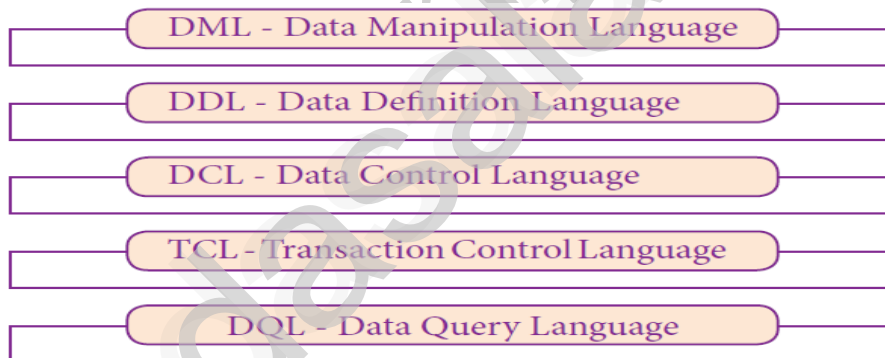
Example:

set_A={'A', 2, 4, 'D'}

set_B={'A', 'B', 'C', 'D'}

print(set_A ^ set_B)

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

---

37   SQL commands are divided into five categories:

DATA DEFINITION LANGUAGE

- The Data Definition Language (DDL) consist of SQL statements used to define the database structure or schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in databases.
- The DDL provides a set of definitions to specify the storage structure and access methods used by the database system.

A DDL performs the following functions : (PTA only DDL performs)

- It should identify the type of data division such as data item, segment, record and database file.
- It gives a unique name to each data item type, record type, file type and data base.
- It should specify the proper data type.
- It should define the size of the data item.
- It may define the range of values that a data item may use.
- It may specify privacy locks for preventing unauthorized data entry.

5

SQL commands which comes under Data Definition Language are:

Create	To create tables in the database.
Alter	Alters the structure of the database.
Drop	Delete tables from database.
Truncate	Remove all records from a table, also release the space occ records.

DATA MANIPULATION LANGUAGE

- A Data Manipulation Language (DML) is a computer programming language used for adding (inserting), removing (deleting), and modifying (updating) data in a database. In SQL, the data manipulation language comprises the SQL-data change statements, which modify stored data but not the schema of the database table.
- After the database schema has been specified and the database has been created, the data can be manipulated using a set of procedures which are expressed by DML.

The DML is basically of two types:

- **Procedural DML –** Requires a user to specify what data is needed and how to get it.
- **Non-Procedural DML** – Requires a user to specify what data is needed without specifying how to get it.

SQL commands which comes under Data Manipulation Language are :

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 o by them.

DATA CONTROL LANGUAGE

- A Data Control Language (DCL) is a programming language used to control the access of data stored in a database. It is used for controlling privileges in the database (Authorization).
- The privileges are required for performing all the database operations such as creating sequences, views of tables etc.
- SQL commands which come under Data Control Language are:

Grant	Grants permission to one or more users to perform specific tasks.
Revoke	Withdraws the access permission given by the GRANT statement.

TRANSACTIONAL CONTROL LANGUAGE

- Transactional control language (TCL) commands are used to manage transactions in the database. These are used to manage the changes made to the data in a table by DML statements.
- SQL command which come under Transfer Control Language are:

Commit	Saves any transaction into the database permanently.
Roll back	Restores the database to last commit state.
Save point	Temporarily save a transaction so that you can rollback.

	DATA QUERY LANGUAGE	
	• The Data Query Language consist of commands used to query or retrieve data from a database. One such SQL command in Data Query Language is	
	Select : It displays the records from the table.	
(OR)	❖ 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 accept an argument of any type, and return multiple values without any kind of declaration beforehand. Whereas in C++ return statement can return only one value.	**5**

38	There are two ways to read a CSV file.
	  1. Use the csv module's reader function
	  2. Use the DictReader class.
	
	CSV Module's Reader Function:
	    • You can read the contents of CSV file with the help of **csv.reader()** method. **The reader function is designed to take each line of the file and make a list of all columns.**
	    • Then, you just choose the column you want the variable data for. Using this method one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (,).
	syntax :
	                csv.reader(fileobject,delimiter,fmtparams)
	  where
	    ❖ **file object :** passes the path and the mode of the file
	    ❖ **delimiter**: an optional parameter containing the standard dilects like , | etc can be omitted
	    ❖ **fmtparams:** optional parameter which help to override the default values of the dialects like skipinitialspace,quoting etc. Can be omitted
	    ❖ CSV file - data with default delimiter comma (,)
	    ❖ CSV file - data with Space at the beginning
	    ❖ CSV file - data with quotes
	    ❖ CSV file - data with custom Delimiters
	CSV file with default delimiter comma (,)
	    ❖ The following program read a file called "sample1.csv" with default delimiter comma (,) and print row by row.
	Program:	

```
import csv
with open('c:\\pyprg\\sample1.csv', 'r') as F:
        reader = csv.reader(F)
        print(row)
F.close()
```

Reading CSV File Into A Dictionary:

- To read a CSV file into a dictionary can be done by using **DictReader** class of csv module which works similar to the reader() class but creates an object which maps data to a dictionary.
- The keys are given by the fieldnames as parameter. **DictReader** works by reading the first line of the CSV and using each comma separated value in this line as a **dictionary key.** The columns in each subsequent row then behave like dictionary values and can be accessed with the appropriate key (i.e. fieldname).
- If the first row of your CSV does not contain your column names, you can pass a fieldnames parameter into the DictReader's constructor to assign the dictionary keys manually.
- The main difference between the csv.reader() and DictReader() is in simple terms csv. reader and csv.writer work with **list/tuple,** while csv.DictReader and csv.DictWriter work with dictionary. csv.DictReader and csv.DictWriter take additional argument fieldnames that are used as dictionary keys.

  Example:
  ```
  import csv
  filename = 'c:\\pyprg\\sample8.csv'
  input_file =csv.DictReader(open(filename,'r'))
  for row in input_file:
  print(dict(row))
  ```
- In the above program, DictReader() is used to read "sample8.csv" file and map into a dictionary. Then, the function dict() is used to print the data in dictionary format without order.
- Remove the dict() function from the above program and use print(row).Check you are getting the following output

  OrderedDict([('ItemName ', 'Keyboard '), ('Quantity', '48')])
  OrderedDict([('ItemName ', 'Monitor'), ('Quantity', '52')])
  OrderedDict([('ItemName ', 'Mouse '), ('Quantity', '20')])

(OR)

- ❖ In the output figure, you can see few buttons at the bottom left corner. Let us see the use of these buttons.
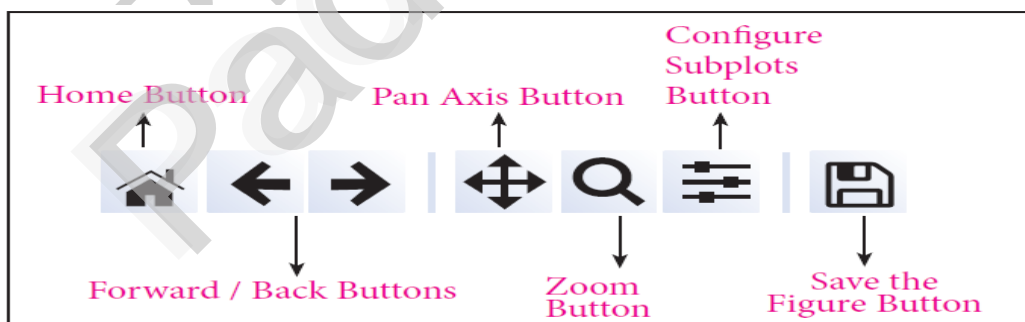
Home Button → The Home Button will help once you have begun navigating your chart. If you ever want to return back to the original view, you can click on this.

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

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

Zoom → The Zoom button lets you click on it, then click and drag a square that you would like to zoom into specifically. Zooming in will require a left click and drag.

5

You can alternatively zoom out with a right click and drag.

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

Save Figure → This button will allow you to save your figure in various forms.

****

Prepared by :

DEPARTMENT OF COMPUTER SCIENCE

M.Mari Muthu :8122169221,8675047607

SHRI VIDHYABHARATHI MATRIC HR.SEC.SCHOOL

SAKKARAMPALAYAM , AGARAM (PO) ELACHIPALAYAM

TIRUCHENGODE(TK), NAMAKKAL (DT) PIN-637202

BE(CSE)
RAGUL S
INDUSTAN ENGINEERING
COLLEGE, COVAI

BE.(AI&DS)
RANJITH D
CIT COLLEGE OF
ENGINEERING COVAI

BE(CSE)
SHARVESH B
VSB ENGINEERING
COLLEGE COVAI

BE(ECE)
SUJITH S
INDUSTAN COLLEGE OF
ENGINEERING, COVAI

BE(CSE)
YUVANSHANKAR S
KONGU ENGINEERING
COLLEGE, PERUNDURAI

BE(Bio-Medical)
POOJA M
KPR ENGINEERING
COLLEGE, COIMBATORE

B.Tech(IT)
SRI HARINI P
KARPAGAM COLLEGE OF
ENGINEERING, COIMBATORE

B.Tech(Bio.Tech)
JEEVITHA G T
VIVEKANANDA COLLEGE OF
ENGINEERING,
ELAYAMPALAYAM

BE(ECE)
KANIKA S
VIVEKANANDA COLLEGE OF
ENGINEERING COLLEGE,
ELAYAMPALAYAM

B.Tech(AI&DS)
SUBHASHREE HARINI N
VELLALAR ENGINEERING
COLLEGE, THINDAL

B.Tech(AGRI)
PRAGADHI C
MANGAIKARASI ENGINEERING
COLLEGE, MADURAI

BE(E&I)
ELAKKIYA R
PERUNDURAI KONGU
ENGINEERING COLLEGE

BE(EEE)
DHARSHINI S
KARPAGAM COLLEGE OF
ENGINEERING, COIMBATORE

BE(EEE)
HARSHINI B
KARPAGAM COLLEGE OF
ENGINEERING, COIMBATORE

BE(ECE)
KANISHKA M P
SNS ENGINEERING
COLLEGE, COIMBATORE

BE(ECE)
MADHU PRIYA P
NGP COLLEGE, COIMBATORE

BE(CS)
MONIKA S
SRM COLLEGE OF
ENGINEERING, TRICHY

BE(CS)
CHARUMATHI A K
JJ COLLEGE OF
ENGINEERING, TRICHY

AIML
SHANCHITHA M
M.KUMARASAMY COLLEGE
OF ENGINEERING, KARUR

BE(ECE)
VARSHA C
KALAINAR KARUNANITHI
INSTITUTE OF TECHNOLOG,
COIMBATORE

B.Tech(AI&DS)
KANISHKA L
KUMARAGURU COLLEGE
OF ENGINEERING, COIMBATORE

BE(CSE)
SONIKA M
KONGU ENGINEERING
COLLEGE, PERUNDURAI

BE(CSE)
SUNITHA B
KARPAKAM INSTITUTE
OF (COIMBATORE)

B.Tech(Bio-Tech)
NITHIKA H
KUMARAGURU COLLEGE OF
ENGINEERING, COIMBATORE

B.Tech(AI&DS)
SINDHUJA R
KONGU ENGINEERING
COLLEGE, PERUNDURAI

B.Tech(IT)
VARSHINI P
KONGU ENGINEERING
COLLEGE, PERUNDURAI

BE(CSE)
AKSHAYA A
AVINASILINGAM,
COIMBATORE

BE(CSE)
BHUMIKA S
SALEM KARUPPOOR
ENGINEERING

2022-2023 ஆம் ஆண்டில் PILOT & BDS & LAW கல்லூரியில் சேர்ந்த மாணவர்கள்

PILET
DEEPAKRAJ R
KERALA,
THIRUVANANDHAPURAM

BA.LLB
CARLINE MARY A
BANGALORE CHRIST UNIVERSITY

BDS
VIDHYA K
NANDHA COLLEGE,
PERUNDURAI

LAW
THARUN A
KALASALINGAM UNIVERSITY,
VIRUTHUNAGAR

B.Pharm
SAKTHIKUMARAN S
COUNCELLING

D.Pharm
SUBASIVAVELAN K
SALEM VINAYAGA MISSION

BA-LLB
MOHITH K
KAVITHA'S LAW
COLLEGE, CHENNAI

CA., B.Com., & B.B.A சேர்ந்த மாணவர்கள்

HARINISRI S	HARSHINI P R	SASMITA A N	JAIPRASATH K	POORNITH R	RAGUVISWANATH M	KAPILAN K	ABINESH M	GAUTAM RAJENDRAN
CA	CA	B.COM	B.COM	B.COM	B.COM	BCA	B.COM	B.COM
KS ACADEMY, CHENNAI	V-S ACADEMY, COIMBATORE	PSG COLLEGE, COIMBATORE	SRI KRISHNA COLLEGE OF ARTS & SCIENCE, COIMBATORE	DR. NGP ARTS & SCIENCE, COIMBATORE	PSG COLLEGE, COIMBATORE	DR. NGP ARTS & SCIENCE, COIMBATORE	SRI KRISHNA COLLEGE OF ARTS & SCIENCE, COIMBATORE	SRI KRISHNA COLLEGE OF ARTS & SCIENCE,

+1 மாணவர்களுக்கான மாநில அளவிலான தமிழ்மொழி இலக்கியத் திறனறித் தேர்வில் தேர்ச்சி பெற்று Rs. 36,000/- ஊக்கத்தொகை பெறுவதற்குத் தேர்வாகியுள்ள ஸ்ரீ வித்யபாரதி சாதனையாளர்கள்

தமிழ்மொழி இலக்கியத் திறனறித் தேர்வு - அக்டோபர் - 2023

ஸ்ரீநிதி பா	கீர்த்திகா வெ	தேவானந்த் ப	முகேஷ்குமார் மு
99/100	99/100	99/100	99/100
STATE 2nd RANK	STATE 2nd RANK	STATE 2nd RANK	STATE 2nd RANK
Reg. No. 223120297	Reg. No. 223120352	Reg. No. 223120274	Reg. No. 223120241

லதிகா இ	ஜீவனிகா சி பு	நிஷாந்த் செ	சம்ரிதா ர	ஸ்ரீஹரிணி மோ	கௌதம் ப	அனிஷ்கா ரா ச	வர்ஷா பெ
98	97	97	97	96	96	95	94
STATE 3rd RANK							
Reg. No. 223120282	Reg. No. 223120345	Reg. No. 223120260	Reg. No. 223120291	Reg. No. 223120316	Reg. No. 223120246	Reg. No. 223120334	Reg. No. 223120310

லோனா ஆக்னஸ் பெ	தநேயா கா	ஹரினிஸ்ரீ மா	சஞ்சய் பாரதி அ சு	வர்ஷினி அ ப	சுபிக்ஷா சு	சன்மதி அ ச	ஸ்ரீஹர்ஷினி மோ
94	94	93	93	93	93	93	92
Reg. No. 223120288	Reg. No. 223120338	Reg. No. 223120340	Reg. No. 223120281	Reg. No. 223120314	Reg. No. 223120306	Reg. No. 223120293	Reg. No. 223120319

தமிழ்மொழி இலக்கியத் திறனறித் தேர்வு - அக்டோபர் - 2022

95/100 தீபிக்ஷா ல த Reg. No. 0551190412	அனுஸ்ரீ த 89/100 Reg. No. 0551190377	திஷ்யா செ 88/100 Reg. No. 0551190415	சரவணன் பா 87/100 Reg. No. 0551190285	சர்வேஸ் கோ த 87/100 Reg. No. 0551190286	சர்விகா சு 85/100 Reg. No. 0551190397
	மிதுன்குமார் வெ ஞா 83/100 Reg. No. 0551190340	ரோகித் ம 81/100 Reg. No. 0551190360	நவினா த 81/100 Reg. No. 0551190417	ராஸ்மிகா பு ரா 81/100 Reg. No. 0551190441	பிரித்திகா மோ ம 80/100 Reg. No. 0551190427