

**12<sup>th</sup>**  
**STD**

**PUBLIC EXAMINATION MARCH - 2023**

Reg. No.

--	--	--	--	--	--	--	--

**COMPUTER SCIENCE (with Answers)**

TIME ALLOWED : 3.00 Hours]

PART - III

[MAXIMUM MARKS : 70

**Instructions :**

- 1) Check the question paper for fairness of printing. If there is any lack of fairness, inform the Hall Supervisor immediately.
- 2) Use **Blue** or **Black** ink to write and underline and pencil to draw diagrams

**PART - I**

**Note :** (i) All the questions. (15×1=15)

(ii) Choose the most appropriate answer from the given **four** alternatives and write the option code and the corresponding answer.

1. \_\_\_\_\_ members are accessible from outside the class.  
(a) Secured members (b) Public members  
(c) Private members (d) Protected members
2. Which of the following is not a keyword in python?  
(a) continue (b) break  
(c) operator (d) while
3. The small sections of code that are used to perform a particular task is called:  
(a) Pseudo code (b) Subroutines  
(c) Modules (d) Files
4. The number of important control structures in python:  
(a) 5 (b) 3 (c) 6 (d) 4
5. Class members are accessed through \_\_\_\_\_ operator.  
(a) # (b) & (c) % (d) .
6. The database Model which represents the Parent-Child relationship:  
(a) Hierarchical (b) Relational  
(c) Object (d) Network
7. The operator which is used for concatenation?  
(a) \* (b) + (c) = (d) &
8. Importing C++ program in a python program is called \_\_\_\_\_.  
(a) Interconnecting (b) Wrapping  
(c) Parsing (d) Downloading
9. \_\_\_\_\_ command is used to remove a table from the database.  
(a) DELETE ALL (b) DROP TABLE  
(c) ALTER TABLE (d) DELETE
10. The function that returns the largest value of the selected column is :  
(a) HIGH () (b) MAX ()  
(c) MAXIMUM () (d) LARGE ()

11. The datatype whose representation is known are called:  
(a) Concrete datatype (b) Built-in datatype  
(c) Abstract datatype (d) Derived datatype
12. A Function which calls itself, is called as:  
(a) Lambda (b) Built-in  
(c) Return statement (d) Recursion
13. The mode which is used when dealing with non-text files like image or exe files:  
(a) xls mode (b) Text mode  
(c) csv mode (d) Binary mode
14. In dynamic programming, the technique of storing the previously calculated values is called:  
(a) Memoization  
(b) Saving value property (c) Mapping  
(d) Storing value property
15. Let set A = {3,6,9}, set B = {1, 3, 9}.  
The result of the following snippet print (set A|set B)  
(a) {1} (b) {3, 6, 9, 1, 3, 9}  
(c) {1, 3, 6, 9} (d) {3, 9}

**PART - II**

**Note :** Answer **any six** questions. Question No. **24** is **compulsory.** **6 × 2 = 12**

16. What is Tuple? Give an example.
17. What is a scope?
18. How will you delete a string in python?
19. Write note on range() in loop.
20. What is class?
21. What is Data Manipulation Language?
22. Mention the default modes of the File.
23. List the general types of data visualization.
24. What will be output of the following python code?  
Squares = [x \*\*2 for x in range (1,11)]  
print (squares)

**PART - III**

**Note :** Answer **any six** questions. Question No. **33** is **compulsory.** **6 × 3 = 18**

25. Mention the characteristics of Interface.
26. What do you understand by Dynamic Programming?

27. Explain Ternary operator with an example.
28. Write the syntax of while loop.
29. Differentiate - ceil () and floor () function.
30. What is the difference between csv reader () method and DictReader () class?
31. Differentiate fetchone () and fetchmany ().
32. Write a Python program to display the given pattern.  
COMPUTER  
COMPUTE  
COMPUT  
COMPU  
COMPU  
COMP  
COM  
CO  
C
33. Write about the steps of Python program executing C++ program using control statement.

**PART - IV****Note :** Answer all the questions:  $5 \times 5 = 25$ 

34. (a) What is a List? Why list, can be called as pairs? Explain with suitable example.  
(OR)  
(b) Discuss about linear search algorithm.
35. (a) Discuss in details about Token in Python.  
(OR)  
(b) Explain the following built-in function.  
(i) id() (ii) chr()  
(iii) round() (iv) type()  
(v) pow()
36. (a) What is Nested Tuple? Explain with an example.  
(OR)  
(b) Explain the different types of relationship mapping.
37. (a) Write the syntax for getotp() and explain its arguments and return values.  
(OR)  
(b) Differentiate DBMS and RDBMS.
38. (a) Explain about differences between Histogram and Bar Graph.  
(OR)  
(b) Explain 'continue' statement with an example.

**ANSWER****PART - I**

1. (b) Public members
2. (c) operator
3. (b) Subroutines
4. (b) 3
5. (d) .

6. (a) Hierarchical
7. (b) +
8. (b) Wrapping
9. (b) DROP TABLE
10. (b) MAX ()
11. (a) Concrete datatype
12. (d) Recursion
13. (d) Binary mode
14. (a) Memoization
15. (a) {1, 3, 6, 9}

**PART - II**

16. (i) A tuple is a comma-separated sequence of values surrounded with parentheses. Tuple is similar to a list.  
(ii) The difference between the two is that you cannot change the elements of a tuple once it is assigned whereas in a list, elements can be changed.  
(iii) **Example :** colour= ('red', 'blue', 'Green')
17. Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.
18. Python will not allow deleting a particular character in a string. Whereas you can remove entire string variable using **del** command.  

```
>>> str1="How about you"
>>> print (str1)
How about you
>>> del str1
>>> print (str1)
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    print (str1)
NameError: name 'str1' is not defined
```
19. 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. Main advantages of functions are
20. (i) Class is the main building block in Python.  
(ii) Class is a template for the object.  
(iii) Object is a collection of data and function that act on those data.  
(iv) Objects are also called as instances of a class.
21. 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.

22. (i) The default is reading ('r') in text mode.  
(ii) In this mode, while reading from the file the data would be in the format of strings.
23. (i) Charts (ii) Tables  
(iii) Graphs (iv) Maps  
(v) Infographics (vi) Dashboards
24. Output  
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

### PART - III

25. (i) The class template specifies the interfaces to enable an object to be created and operated properly.  
(ii) An object's attributes and behaviour is controlled by sending functions to the object.
26. (i) Dynamic programming is an algorithmic design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions.  
(ii) Dynamic programming approach is similar to divide and conquer. The given problem is divided into smaller and yet smaller possible sub-problems.  
(iii) Dynamic programming is used whenever problems can be divided into similar sub-problems. So that their results can be re-used to complete the process.  
(iv) Dynamic programming approaches are used to find the solution in optimized way. For every inner sub problem, dynamic algorithm will try to check the results of the previously solved sub-problems. The solutions of overlapped sub-problems are combined in order to get the better solution.
27. (i) Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false.  
(ii) It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

#### Syntax :

Variable Name = [on\_true] if [Test expression] else [on\_false]

#### (iii) Example:

```
min = 49 if 49 < 50 else 50 // min = 49
min = 50 if 49 > 50 else 49 // min = 49
```

#### 28. Syntax:

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

#### 29.

S.No	ceil ()	floor ()
(i)	Returns the smallest integer greater than or equal to x.	Returns the largest integer less than or equal to x.
(ii)	Syntax : math.ceil(x)	Syntax : math.floor(x)

#### 30. Reader():

- (i) The reader function is designed to take each line of the file and make a list of all columns.  
(ii) Using this function one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (.).  
(iii) csv. Reader work with list/tuple.  
(iv) Syntax : csv.reader(fileobject,delimiter, fmtparams)

#### DictReader() :

- (i) DictReader works by reading the first line of the CSV and using each column comma separated value in this line as a dictionary key.  
(ii) DictReader is a class of csv module is used to read a CSV file into a dictionary.  
(iii) It creates an object which maps data to a dictionary.  
(iv) csv.DictReader work with dictionary.

#### 31.

	fetchone()	fetchmany()
(i)	The fetchone() method returns the next row of a query result set or None in case there is no row left	The fetchmany() method returns the next number of rows (n) of the result set.
(ii)	Using while loop and fetchone() method we can display all the records from a table.	Displaying specified number of records is done by using fetchmany().

#### 32. str1 = "COMPUTER"

```
index = len(str1)
for i in str1:
    print (str1[: index])
    index - = 1
```

33. The steps for executing the C++ program to check a given number is palindrome or not is given below  
Step 1 : Type the C++ program to check whether the input number is palindrome or not in notepad and save it as "pali\_cpp.cpp".  
Step 2 : Type the Python program and save it as pali.py

Step 3 : Click the Run Terminal and open the command window

Step 4 : Type the command Python  
pali.py -i pali\_cpp

### PART - IV

34. (a)

**List :**

(i) List is constructed by placing expressions within square brackets separated by commas. Such an expression is called a list literal. List can store multiple values. Each value can be of any type and can even be another list. Example for List is [10, 20].

(ii) 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.

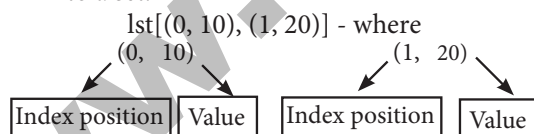
```
lst := [10, 20]
x, y := lst
```

(iii) In the above example x will become 10 and y will become 20. A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.

(iv) Unlike a list literal, a square-brackets expression directly following another expression does not evaluate to a list value, but instead selects an element from the value of the preceding expression.

```
lst[0]
10
lst[1]
20
```

(v) In both the example mentioned above mathematically we can represent list similar to a set.



**Pair :**

Any way of bundling two values together into one can be considered as a pair. Lists are a common method to do so. Therefore List can be called as Pairs.

(OR)

- (b) (i) Linear search also called sequential search is a sequential method for finding a particular value in a list.
- (ii) This method checks the search element with each element in sequence until the desired element is found or the list is exhausted. In

this searching algorithm, list need not be ordered.

**Pseudo code :**

- (i) Traverse the array using for loop
- (ii) In every iteration, compare the target search key value with the current value of the list.
  - If the values match, display the current index and value of the array
  - If the values do not match, move on to the next array element.
- (iii) If no match is found, display the search element not found.

**Example**

To search the number 25 in the array given below, linear search will go step by step in a sequential order starting from the first element in the given array if the search element is found that index is returned otherwise the search is continued till the last index of the array. In this example number 25 is found at index number 3.

index	0	1	2	3	4
values	10	12	20	25	30

**Example 1 :**

Input: values[] = {5, 34, 65, 12, 77, 35}

target = 77

**Output:** 4

**Example 2:**

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}

target = 200

**Output:** -1 (not found)

35. (a)

Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**. The normal token types are

- (i) Identifiers,
- (ii) Keywords,
- (iii) Operators,
- (iv) Delimiters and
- (v) Literals.

(i) **Identifiers :**

- An Identifier is a name used to identify a variable, function, class, module or object.
- An identifier must start with an alphabet (A..Z or a..z) or underscore ( \_ ).
- Identifiers may contain digits (0 .. 9).
- Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
- Identifiers must not be a python keyword.
- Python does not allow punctuation character such as %, \$, @ etc., within identifiers.

**Example :**

**Example of valid identifiers :** Sum, total\_marks, regno, num1

**Example of invalid identifiers :** 12Name, name\$, total-mark, continue



(ii) **Keywords :**

- Keywords are special words used by Python interpreter to recognize the structure of program..
- As these words have specific meaning for interpreter, they cannot be used for any other purpose.
- Python keywords :** false, class, If, elif, else, pass, break etc.

(iii) **Operators :**

- In computer programming languages operators are special symbols which represent computations, conditional matching etc.
- The value of an operator used is called **operands**.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment etc. Value and variables when used with operator are known as **operands**.

**Example :**

```
a=100
b=10
print ("The sum = ",a+b)
print ("The a > b =", a>b)
print ("The a > b or a == b =", a > b or
a==b)
a+=10
print("The a+=10 is =", a)
```

**Output :**

```
The sum = 110
The a>b = True
The a > b or a == b = True
The a+=10 is = 110
```

- (iv) **Delimiters :** Python uses the symbols and symbol combinations as delimiters in expressions, lists, dictionaries and strings. Following are the delimiters.

(	)	[	]	{	}
,	:	.	'	=	;
+=	-=	*=	/=	//=	%=
&=	=	^=	>>=	<<=	**=

- (v) **Literals :** Literal is a raw data given to a variable or constant. In Python, there are various types of literals.

- Numeric Literals** consists of characters surrounded by quotes.
- String literal** is a sequence of characters surrounded by quotes.
- Boolean literal** can have any of the two values : True or False.

(OR)

## (b) (a)

Function	Description	Syntax	Example
id ()	id() Return the "identity" of an object. i.e. the address of the object in memory. <b>Note:</b> The address of x and y may differ in your system.	id (object)	x=15 y='a' print ('address of x is ',id (x)) print ('address of y is ',id (y)) <b>Output:</b> address of x is : 1357486752 address of y is : 13480736

## (b)

Function	Description	Syntax	Example
chr ()	Returns the Unicode character for the given ASCII value. This function is inverse of ord() function.	chr (i)	c=65 d=43 print (chr (c)) print (chr (d)) <b>Output:</b> A +

## (c)

Function	Description	Syntax	Example
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 y= 22.2 z= -18.3 print ('x value is rounded to', round (x)) print ('y value is rounded to', round (y)) print ('z value is rounded to', round (z))

(d)

Function	Description	Syntax	Example
type ( )	Returns the type of object for the given single object. <b>Note:</b> This function used with single object parameter.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) <b>Output:</b> <class 'float'> <class 'str'> <class 'bool'>

(e)

Function	Description	Syntax	Example
pow ( )	Returns the computation of ab i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) <b>Output:</b> 25 125.0 343

36. (a)

**Nested Tuples :**

(i) 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.

(ii) **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)

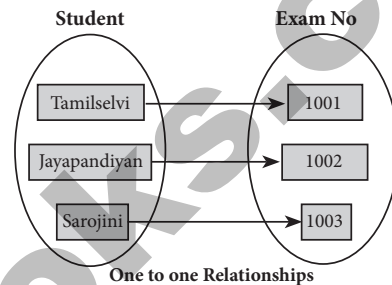
(iii) **Output :**

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

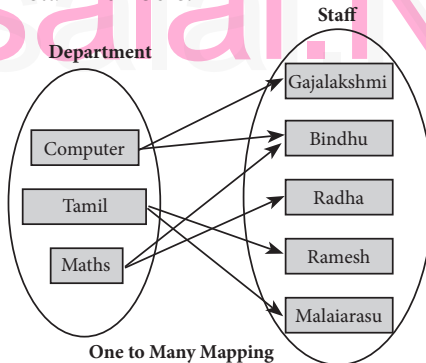
(OR)

(b) The types of relationships used in a database.

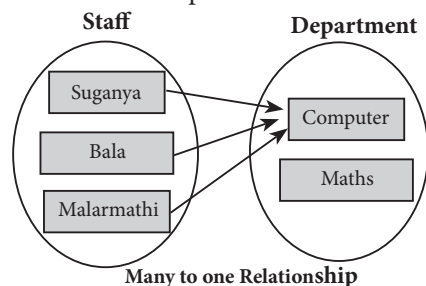
- (i) One-to-One Relationship
  - (ii) One-to-Many Relationship
  - (iii) Many-to-One Relationship
  - (iv) Many-to-Many Relationship
- (i) **One-to-One Relationship :** In One-to-One Relationship, one entity is related with only one other entity. One row in a table is linked with only one row in another table and vice versa.

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

- (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.

- (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.



(iv) **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 1 : Customers and Product**

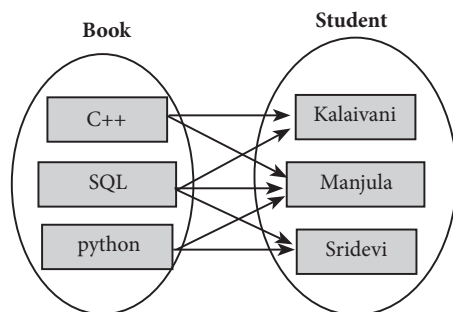
Customers can purchase various products and Products can be purchased by many customers

▪ **Example 2 : Students and Courses**

A student can register for many Courses and a Course may include many students

▪ **Example 3 : Books and Student.**

Many Books in a Library are issued to many students.



Many to Many Relationship

37. (a)

**Python getopt Module :**

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

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

**getopt.getopt method** : This method parses command-line options and parameter list. Following is the syntax for this method –  
`<opts>,<args>=getopt.getopt(argv, options, [long_options])`

Here is the detail of the parameters –

(i) **argv** : This is the argument list of values to be parsed (splited). In our program the complete command will be passed as a list.

(ii) **options** : This is string of option letters that the Python program recognize as, for input or for output, with options (like 'i' or 'o') that followed by a colon (:). Here colon is used to denote the mode.

(iii) **long\_options** : This contains a list of strings. Argument of Long options should be followed by an equal sign ('='). In our program the C++ file name along with its path will be passed as string and 'i' will be also passed to indicate it as the input file.

**getopt() method returns value consisting of two elements.** Each of these values are stored separately in two different list (arrays) **opts** and **args**. Opts contains list of splitted strings like mode and path. args contains error string, if at all the comment is given with wrong path or mode. args will be an empty list if there is no error.

**For example**, The Python code which is going to execute the C++ file p4 in command line will have the getopt() method like the following one.

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

where <b>opts</b> contains	<code>[('-i', 'c:\\pyprg\\p4')]</code>
<code>-i :-</code>	<b>option</b> nothing but <b>mode should be followed by :</b>
<code>'c:\\pyprg\\p4'</code>	<b>value</b> nothing but the <b>absolute path of C++ file.</b>

In our examples since the entire command line commands are parsed and no leftover argument, the **second argument args** will be empty []. If args is displayed using print() command it displays the output as [].

(OR)

(b)

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.
Example	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite.

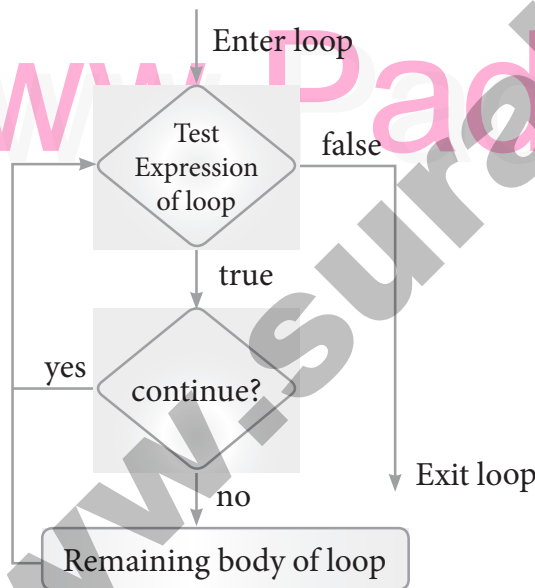
38. (a)

	Histogram	Bar graph
(i)	Histogram refers to a graphical representation; that displays data by way of bars to show the frequency of numerical data.	A bar graph is a pictorial representation of data that uses bars to compare different categories of data.
(ii)	A histogram represents the frequency distribution of continuous variables.	Conversely, a bar graph is a diagrammatic comparison of discrete variables.
(iii)	Histogram presents numerical data.	Bar graph shows categorical data.
(iv)	Items of the histogram are numbers, which are categorised together, to represent ranges of data.	As opposed to the bar graph, items are considered as individual entities.
(v)	A histogram, this cannot be done, as they are shown in the sequence of classes.	In the case of a bar graph, it is quite common to rearrange the blocks, from highest to lowest.

(OR)

(b) Continue statement unlike the break statement is used to skip the remaining part of a loop and start with next iteration.

**Syntax :**  
continue



Working of continue statement

The working of continue statement in for and while loop is shown below.

**for var in sequence:**

# code inside for loop

**if condition:**

continue

#code inside for loop

#code outside for loop

**while test expression:**

#code inside while loop

**if condition:**

continue

#code inside while loop

#code outside while loop

**Example:**

Program to illustrate the use of continue statement inside for loop

```
for word in "Jump Statement":
```

```
    if word == "e":
```

```
        continue
```

```
    print (word, end = ' ')
```

```
print ("\n End of the program")
```

**Output:**

Jump Statement

End of the program

The above program is same as the program we had written for 'break' statement except that we have replaced it with 'continue'.

