12th STD

PUBLIC EXAMINATION - MARCH 2025

PART - III

Reg. No.

TIME ALLOWED : 3.00 Hours] **COMPUTER SCIENCE (with Answers)** [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) **Answer 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. Which of the following is a distinct syntactic block?
(a) Subroutines (b) Function
(c) Definition (d) Modules

2. Which of the following functions that build the abstract data type?
(a) Constructors (b) Recursive
(c) Destructors (d) Nested

3. The _____ rule is used to decide the order in which the scopes are to be searched.
(a) LBEG (b) LEGB
(c) GEBL (d) GBLE

4. Binary search is also called as_____.
(a) Random search (b) Linear search
(c) Half-interval search (d) Sequential search

5. Which operator is also called as conditional operator?
(a) Logical (b) Ternary
(c) Assignment (d) Relational

6. Which of the following is known as definite loop?
(a) for (b) do..while
(c) if..else (d) while

7. Which of the following keyword is used to exit a function block?
(a) finally (b) define
(c) def (d) return

8. The subscript of a string may be :
(a) positive or negative numbers
(b) positive numbers
(c) both positive and negative numbers
(d) negative numbers

9. Which of the following is the output of the given Python code?
```
str1="Computer Science"
print(str1[ : :–1])
```
(a) ecneicS retupmoC
(b) Computer Science
(c) eniS eumC
(d) Cmue Sine

10. Class members are accessed through which operator?
(a) # (b) & (c) % (d) .

11. Who developed ER model?
(a) Chend (b) Chen
(c) Chand (d) E F Codd

12. Queries can be generated using :
(a) MODIFY (b) SELECT
(c) ALTER (d) ORDER BY

13. Making some changes in the data of the existing file or adding more data is called :
(a) Modification (b) Editing
(c) Alteration (d) Appending

14. The module which allows to interface with the Windows operating system is :
(a) csv module (b) OS module
(c) getopt module (d) sys module

15. Which of the following keyword avoids the duplicate?
(a) Where (b) Distinct
(c) GroupBy (d) Remove

PART - II

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

16. What is a pair? Give an example.

17. What is sorting?

18. What is the output of the following Python code?
```
for x in range (5, 20, 5) :
    print (x, end = ' ')
```

19. What is Slicing?

20. What is the purpose of Destructor?

21. Which component of SQL lets insert values in tables and which lets to create a table?

22. What is the use of cd command? Give an example.

23. How will you install matplotlib?

24. What is the output of the following Python code?
```
strl = 'WELCOME'
print(strl.islower ( ) )
print(strl.lower ( ) )
```

PART - III

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

25. What are the side effects of Impure function? Give an example.

26. Define Global scope with an example.

27. What are the assignment operators that can be used in Python?

28. Differentiate ceil() and floor () functions.

29. Write the syntax of the following Python list functions. (i) remove() (ii) pop( ) (iii) clear( )

30. Explain Cartesian product with a suitable example.

31. Write the difference between reader() and DictReader().

32. Mention the difference between fetchone() and fetchmany().

33. What will be the output of the following Python code?

```
squares = []
for x in range(1,6) :
    s=x**2
    squares.append(s)
print(squares)
```

PART - IV

Note : Answer **all the questions:** **5 × 5 = 25**

34. (a) Explain pure and impure functions with an example. **(OR)**

(b) Explain the different types of data model.

35. (a) Explain the different types of operators in Python. **(OR)**

(b) Write the different types of constraints and their functions.

36. (a) Explain While loop in Python with an example. **(OR)**

(b) Tabulate the different Python file modes with description.

37. (a) Explain the recursive function with an example. **(OR)**

(b) What is the purpose of sys, os, getopt, module in Python? Explain.

38. (a) Write the output of the following.

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
print(A|B)
print(A&B)
print(A–B)
print(B–A)
print(A∧B)
```

(OR)

(b) Explain the following SQL aggregate functions with suitable Python statement with SQL query.

(i) AVG() (ii) COUNT()
(iii) SUM() (iv) MAX()
(v) MIN()

ANSWER

PART - I

1. (c) Definition **2.** (a) Constructors
3. (b) LEGB
4. (c) Half-interval search
5. (b) Ternary **6.** (a) for
7. (d) return
8. (a) positive or negative numbers
9. (a) ecneicS retupmoC
10. (d) . **11.** (b) Chen
12. (b) SELECT **13.** (a) Modification
14. (b) OS module **15.** (b) Distinct

PART - II

16. 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. **Example :** lst = [(0,10), (1,20)]

17. Sorting is a method of arranging group of items in ascending or descending order. Various sorting techniques in algorithms are Bubble sort, Quick sort, Heap sort, Selection sort, Insertion sort.

18. **Output :**

5
10
5

19. (i) Slice is a substring of a main string. A substring can be taken from the original string by using [] operator and index or subscript values.

(ii) Thus, [] is also known as slicing operator. Using slice operator, you have to slice one or more substrings from a main string.

20. Destructor is also a special method to destroy the objects. In Python, _ _del_ _( ) method is used as destructor. It is just opposite to constructor.

21.

Command	Description	Component
Insert	Inserts data into a table	DML
Create	To create tables in the database	DDL

22. 'cd' command refers to change directory and absolute path refers to the complete path where python is installed. **(Eg)** "cd:\>cd c:\program files\ open office 4\ program"

23. (i) Matpotlib can be installed using pip software.

(ii) Pip is a package manager software for installing python packages.

(iii) After installing Matplotlib, we will begin coding by importing Matplotlib using the command: import matplotlib.pyplot as plt.

24. False.
welcome

PART - III

25. The variables used inside the function may cause side effects though the functions which are not passed with any arguments. In such cases the function is called impure function.
For example the mathematical function random() will give different outputs for the same function call.

```
let randomnumber :=
a := random()
if a > 10 then
return: a
else
return: 10
```

26. (i) A variable which is declared outside of all the functions in a program is known as Global variable.

(ii) This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

1. a:=10 2. Disp(): 3. a:=7 4. print a 5. Disp() 6. print a	Entire program a:=10 Disp( ): a:=7 print a Disp ( ): print a	Output of the Program 7 10

(iii) On execution of the above code the variable a which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because 'a' is defined in global scope.

27. (i) In Python, = is a simple assignment operator to assign values to variable. Let **a** = 5 and **b** = 10 assigns the value 5 to **a** and 10 to **b** these two assignment statement can also be given as **a,b=5,10** that assigns the value 5 and 10 on the right to the variables a and b respectively.

(ii) There are various compound operators in Python like +=, -=, *=, /=, %=, **= and //= are also available.

28.

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)

29. (i) **remove ( ) :** List.remove(element) # to delete a particular element

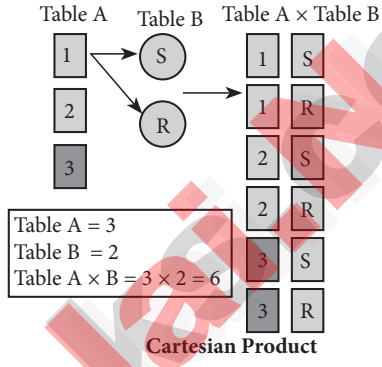
(ii) **pop ( ) :** List.pop(index of an element)

(iii) **clear ( ) :** List.clear( )

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

(ii) A × B means A times B, where the relation A and B have different attributes.

(iii) This type of operation is helpful to merge columns from two relations.



Cartesian Product

31. **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 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.

32.

	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().
	Example: r = cursor.fetchone()	Example: r = cursor.fetchmany()

33. **Output :** [1,4,9,16,25]

PART - IV

34. (a) Pure functions :

(i) Pure functions are functions which will give exact result when the same arguments are passed.

(ii) For example the mathematical function sin (0) always results 0. This means that every time you call the function with the same arguments, you will always get the same result.

(iii) A function can be a pure function provided it should not have any external variable which will alter the behaviour of that variable.

Let us see an example

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

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

(v) There are various theoretical advantages of having pure functions. One advantage is that if a function is pure, then if it is called several times with the same arguments, the compiler only needs to actually call the function once.

Example :

```
let length s:=
    i: = 0
    let i: = 0;
    if i <strlen (s) then
    -- Do something which doesn't affect s
    ++i
```

(vi) If it is compiled, strlen (s) is called each time and strlen needs to iterate over the whole of 's'. If the compiler is smart enough to work out that strlen is a pure function and that 's' is not updated in the loop, then it can remove the redundant extra calls to strlen and make the loop to execute only one time.

(vii) From these what we can understand, strlen is a pure function because the function takes one variable as a parameter, and accesses it to find its length. This function reads external memory but does not change it, and the value returned derives from the external memory accessed.

Impure functions :

(i) The variables used inside the function may cause side effects though the functions which are not passed with any arguments. In such cases the function is called impure function.

(ii) When a function depends on variables or functions outside of its definition block, you can never be sure that the function will behave the same every time it's called. For example the mathematical function random() will give different outputs for the same function call.

```
let randomnumber:=
    a := random()
        if a > 10 then
            return: a
else
            return: 10
```

(iii) Here the function Random is impure as it is not sure what will be the result when we call the function.

(OR)

(b) The different types of a Data Model: Hierarchical Model, Relational Model, Network Database Model, Entity Relationship Model, Object Model.

I. Hierarchical Model :

(i) Hierarchical model was developed by IBM as Information Management System. In Hierarchical model, data is represented as a simple tree like structure form.

(ii) This model represents a one-to-many relationship i.e., parent-child relationship. One child can have only one parent but one parent can have many children. This model is mainly used in IBM Main Frame computers.

II. Relational Model :

(i) The Relational Database model was first proposed by E.F. Codd in 1970 . Nowadays, it is the most widespread data model used for database applications around the world.

(ii) The basic structure of data in relational model is tables (relations). All the information's related to a particular type is stored in rows of that table.

(iii) Hence tables are also known as relations in a relational model. A relation key is an attribute which uniquely identifies a particular tuple (row in a relation (table)).

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

(i) In hierarchical model, a child record has only one parent node,

(ii) In a Network model, a child may have many parent nodes. It represents the data in many-to-many relationships.

(iii) This model is easier and faster to access the data.

IV. Entity Relationship Model. (ER model) :

(i) In this database model, relationship are created by dividing the object into entity and its characteristics into attributes.

(ii) It was developed by Chen in 1976. This model is useful in developing a conceptual design for the database. It is very simple

and easy to design logical view of data. The developer can easily understand the system by looking at ER model constructed.

V. **Object Model :**

(i) Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.

(ii) This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.

(iii) It is used in file Management System. It represents real world objects, attributes and behaviors. It provides a clear modular structure. It is easy to maintain and modify the existing code.

35.

(a) 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.

(i) **Arithmetic operators:** An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic. Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.

(ii) **Relational or Comparative operators:** A Relational operator is also called as Comparative operator which checks the relationship between two operands. If the relation is true, it returns True; otherwise it returns False.

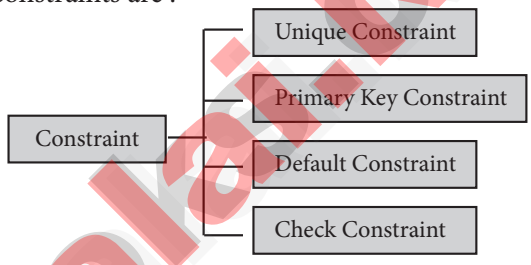
(iii) **Logical operators:** In python, Logical operators are used to perform logical operations on the given relational expressions. There are three logical operators they are and, or and not.

(iv) **Assignment operators :** In Python, = is a simple assignment operator to assign values to variable. Let a = 5 and b = 10 assigns the value 5 to a and 10 to b these two assignment statement can also be given as a,b=5,10 that assigns the value 5 and 10 on the right to the variables a and b respectively. There are various compound operators in Python like +=, -=, *=, /=, %=, **= and //= are also available.

(v) **Conditional operator :** Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false. It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

(OR)

(b) **Type of Constraints :** Constraints ensure database integrity, therefore known as database integrity constraints. The different type of constraints are :



Unique Constraint :

(i) This constraint ensures that no two rows have the same value in the specified columns. For example UNIQUE constraint applied on Admno of student table ensures that no two students have the same admission number and the constraint can be used as:

(ii) **CREATE TABLE Student :**

```
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
Age integer,
Place char (10),
);
```

(iii) The **UNIQUE** constraint can be applied only to fields that have also been declared as **NOT NULL**.

Primary Key Constraint :

(i) This constraint declares a field as a Primary key which helps to uniquely identify a record. It is similar to unique constraint except that only one field of a table can be set as primary key.

(ii) The primary key does not allow **NULL** values.) Example showing Primary Key Constraint in the student table:

CREATE TABLE Student :

```
(
Admno integer PRIMARY KEY, → Primary Key constraint
```

```
Name char(20)NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

(iv) In the above example the Admno field has been set as primary key and therefore will help us to uniquely identify a record, it is also set NOT NULL, therefore this field value cannot be empty.

Default Constraint :

(i) The DEFAULT constraint is used to assign a default value for the field. When no value is given for the specified field having DEFAULT constraint, automatically the default value will be assigned to the field.

(ii) Example showing DEFAULT Constraint in the student table:

```
CREATE TABLE Student :
(
Admno integer PRIMARY KEY,
Name char(20)NOT NULL,
Gender char(1),
Age integer DEFAULT 17,
                    → Default Constraint
Place char(10)
);
```

(i) In the above example the "Age" field is assigned a default value of 17, therefore when no value is entered in age by the user, it automatically assigns 17 to Age.

Check Constraint :

(i) This constraint helps to set a limit value placed for a field. When we define a check constraint on a single column, it allows only the restricted values on that field.

(ii) Example showing check constraint in the student table:

```
CREATE TABLE Student :
(
Admno integer PRIMARY KEY
Name char(20)NOT NULL,
Gender char(1),
Age integer (CHECK<=19), → Check Constraint
Place char(10),
);
```

In the above example the check constraint is set to Age field where the value of Age must be less than or equal to 19.

Table Constraint :

(i) When the constraint is applied to a group of fields of the table, it is known as Table constraint. The table constraint is normally given at the end of the table definition.

(ii) Let us take a new table namely Student1 with the following fields Admno, Firstname, Lastname, Gender, Age, Place:

```
CREATE TABLE Student 1 :
(
Admno integer NOT NULL,
Firstname char(20),
Lastname char(20),
Gender char(1),
Age integer,
Place char(10),
PRIMARY KEY (Firstname, Lastname) →
                        Table constraint
);
```

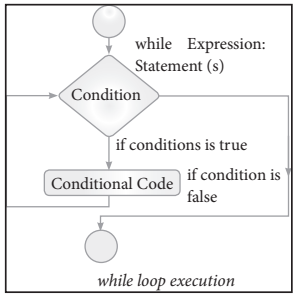
In the above example, the two fields, Firstname and Lastname are defined as Primary key which is a Table constraint.

36. (a) while loop

(i) The syntax of while loop in Python has the following syntax:

Syntax:

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



(ii) In the while loop, the condition is any valid Boolean expression returning True or False. The else part of while is optional part of while. The statements block1 is kept executed till the condition is True. If the else part is written, it is executed when the condition is tested False. Recall while loop belongs to entry check loop type, that is it is not executed even once if the condition is tested False in the beginning. **Eg :** program to illustrate the use of while loop - to print all numbers from 10 to 15

```
i=10                          # initializing part of
                                the control variable
while (i<=15):                # test condition
      print (i,end='\t')      # statements - block 1
      i=i+1                   # Updation of the
                                control variable
```

Output :

10 11 12 13 14 15

(OR)

(b)

Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

37.

(a) (i) When a function calls itself is known as recursion.

(ii) Recursion works like loop but sometimes it makes more sense to use recursion than loop.

(iii) You can convert any loop to recursion. A recursive function calls itself.

(iv) Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.

(v) The condition that is applied in any recursive function is known as base condition.

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

Overview of how recursive function works :

(i) Recursive function is called by some external code.

(ii) If the base condition is met then the program gives meaningful output and exits.

(iii) Otherwise, function does some required processing and then calls itself to continue recursion.

Here is an example of recursive function used to calculate factorial.

Example :

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

Output: 120

(OR)

(b) **(1) Python's sys module :** sys module provides access to some variables used by the interpreter and to functions that interact with the interpreter

sys.argv :

(i) sys.argv is the list of command-line arguments passed to the Python program. **argv contains** all the items that come via the command-line input, it's basically a list holding the command-line arguments of the program.

(ii) To use sys.argv, import sys should be used. The first argument, sys.argv[0] contains the name of the python program (example pali.py) and sys.argv [1]is the next argument passed to the program (here it is the C++ file), which will be the argument passed through main ().

(2) Python's OS Module :

(i) The **OS** module in Python provides a way of using operating system dependent functionality.

(ii) The functions that the **OS** module allows you to interface with the Windows operating system where Python is running on.

os.system():

(i) Execute the C++ compiling command (a string contains Unix, C command which also supports C++ command) in the shell (Here it is Command Window).

(ii) **Command :** os.system ('g++' + <varaiable_ name1> '-<mode>' + <variable_name2>

(3) 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 function :

(i) This function parses command-line options and parameter list. Following is the syntax for this method –

(ii) <opts>,<args>=getopt.getopt(argv, options, [long_options])

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

- **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.
- **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' i will be also passed to indicate it as the input file.

(iii) **getopt()** method returns value consisting of two elements.

(iv) Each of these values are stored separately in two different list (arrays) opts and args.

(v) **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.

(vi) **args** will be an empty list if there is no error.

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

38. (a)

Output:

```
print(A | B)  - {1, 2, 3, 4, 5, 6, 7, 8}
print(A & B) - {4, 5}
print(A - B) - {1, 2, 3}
print(B - A) - {6, 7, 8}
print(A ^ B) - {1, 2, 3, 6, 7, 8}
```

(OR)

(b) (i) AVG() :

The following SQL statement in the python program finds the average mark of all students.

Example :

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT AVG(AVERAGE) FROM student ")
result = cursor.fetchall()
print(result)
```

Output :

```
[(84.65714285714286,)]
```

(ii) COUNT() function :

The SQL COUNT() function returns the number of rows in a table satisfying the criteria specified in the WHERE clause. COUNT() returns 0 if there were no matching rows.

Example : In this example we are going to count the number of records(rows)

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(*) FROM student ")
result = cursor.fetchall()
print(result)
```

Output:

```
[(7,)]
```

(iii) SUM():

The following SQL statement in the python program finds the sum of all average in the Average field of "Student table".

Example :

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT SUM(AVERAGE) FROM student ")
result = cursor.fetchall()
print(result)
```

Output :

```
[(592.6,)]
```

(iv) MAX() AND MIN() FUNCTIONS :

(a) The MAX() function returns the largest value of the selected column.

(b) The MIN() function returns the smallest value of the selected column.

(c) The following example show the highest and least average student's name.

Example :

```
import sqlite3
connection = sqlite3.
connect("Organization.db")
cursor = connection.cursor()
print("Displaying the name of the Highest Average")
cursor.execute("SELECT
 sname,max(AVERAGE) FROM student ")
result = cursor.fetchall()
print(result)
print("Displaying the name of the Least Average")
cursor.execute("SELECT
 sname,min(AVERAGE) FROM student ")
result = cursor.fetchall()
print(result)
```

Output :

```
Displaying the name of the Highest Average
[('PRIYA', 98.6)]
Displaying the name of the Least Average
[('TARUN', 62.3)]
```

❁❁❁❁❁