

12th STD

PUBLIC EXAM MAY - 2022 COMPUTER SCIENCE (with Answers)

Reg. No.

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

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) 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) Definition
 - (b) Subroutines
 - (c) Modules
 - (d) Function
2. Which of the following will retrieve information from the data type?
 - (a) Recursive
 - (b) Constructors
 - (c) Nested
 - (d) Selectors
3. Containers for mapping names of variables to object is called:
 - (a) Binding
 - (b) Scope
 - (c) Namespaces
 - (d) Mapping
4. The word comes from the name of a Persian Mathematician Abu Jafar Mohammed ibn-i Musa al Khowarizmi is called :
 - (a) Algorithm
 - (b) Flowchart
 - (c) Syntax
 - (d) Flow
5. Which Operator is also called as Comparative Operator?
 - (a) Logical Operator
 - (b) Arithmetic Operator
 - (c) Assignment Operator
 - (d) Relational Operator
6. 'elif' can be considered to be short form of _____.
 - (a) else if
 - (b) nested if
 - (c) if...elif
 - (d) if...else
7. Which function is called as anonymous un-named function?
 - (a) Recursion
 - (b) Lambda
 - (c) Define
 - (d) Built-in
8. What will be the output of the following code?


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

 - (a) Type Error
 - (b) Chennai-Schools
 - (c) Chennai
 - (d) Chenna-School
9. If List = [17, 23, 41, 10] then , List.append(32) will result :
 - (a) [10, 17, 23, 32, 41]
 - (b) [32, 17, 23, 41, 10]
 - (c) [41, 32, 23, 17, 10]
 - (d) [17, 23, 41, 10, 32]
10. Which of the following method is used as destructor?
 - (a) __rem__()
 - (b) __init__()
 - (c) __del__()
 - (d) __dest__()
11. A tuple is also known as :
 - (a) Attribute
 - (b) Table
 - (c) Field
 - (d) Row
12. The clause used to sort data in a database :
 - (a) GROUP BY
 - (b) SORT BY
 - (c) SELECT
 - (d) OREDR BY
13. The expansion of CRLF is:
 - (a) Control Router and Line Feed
 - (b) Control Return and Line Feed
 - (c) Carriage Return and Line Feed
 - (d) Carriage Return and Form Feed
14. A Framework for interfacing Python and C++ is :
 - (a) Cython
 - (b) Ctypes
 - (c) Boost
 - (d) SWIG
15. SQLite falls under which database system?
 - (a) Hierarchical Database System
 - (b) Flat File Database System
 - (c) Object Oriented Database System
 - (d) Relational Database System

[1]

PART - II

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

16. What is Abstract Data Type?
17. What is Mapping?
18. What is Searching? Write its types.
19. What are the different modes that can be used to test Python Program?
20. Write the syntax of creating a tuple with 'n' number of elements.
21. Differentiate unique constraint and primary key constraint.
22. What is CSV File?
23. Define Data Visualization.
24. Write the syntax of getopt.getopt method.

PART - III

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

25. List the characteristics of an Algorithm.
26. List the difference between break and continue statements.
27. Write the rules of Local variable.
28. What is sys.argv?
29. What is the use of 'where clause'? Give a Python statement using the where clause.
30. Write any three uses of data visualization.
31. Write short notes on Arithmetic Operator with example.
32. Write a SQL statement using DISTINCT keyword.
33. Write a program to get the following output:

A				
A	B			
A	B	C		
A	B	C	D	
A	B	C	D	E

PART - IV

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

34. (a) (i) What are called Parameters?
(ii) Write a note on: Parameter without type and Parameter with type.

(OR)

(b) Explain LEGB rule with example.

35. (a) Explain Bubble Sort Algorithm with example.

(OR)

(b) Construct the following SQL statements in the student table.

- (i) SELECT statement using GROUP BY Clause

(ii) SELECT statement using ORDER BY Clause

36. (a) Explain input() and output() functions with example.

(OR)

(b) Explain the purpose of the following functions:

- (i) plt.xlabel (ii) plt.ylabel
- (iii) plt.title (iv) plt.legend()
- (v) plt.show()

37. (a) Write a detail note on for loop.

(OR)

(b) Differentiate Excel file and CSV file.

38. (a) Example the different set operations supported by Python with suitable example.

(OR)(b) Explain each word of the following command.
Python <filename.py> - <i> <C++ filename without cpp extension>**ANSWER****PART - I**

1. (a) Definition
2. (d) Selectors
3. (c) Namespaces
4. (a) Algorithm
5. (d) Relational Operator
6. (a) else if
7. (b) Lambda
8. (a) Type Error
9. (d) [17, 23, 41, 10, 32]
10. (c) `___del___()`
11. (d) Row
12. (d) ORDER BY
13. (c) Carriage Return and Life Feed
14. (c) Boost
15. (d) Relational Database System

PART - II

16. **Abstract Data Type:**

- (i) 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.
- (ii) The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.

17. The process of binding a variable name with an object is called mapping. = (equal to sign) is used in programming languages to map the variable and object.
18. A searching algorithm is the step-by-step procedure used to locate specific data among a collection of data. There are two type of searching are
(i) Linear Search (ii) Binary Search
19. (i) In Python, programs can be written in two namely Interactive mode and Script mode.
(ii) Interactive mode allows us to write codes in Python command prompt (>>>).
(iii) Script mode is used to create and edit python source file with the extension .py
20. **Syntax:**
 Tuple_Name = (E1, E2, E2 En) # Tuple with n number elements
 Tuple_Name = E1, E2, E3 En # Elements of a tuple without parenthesis

21.

	Unique Key Constraint	Primary Key Constraint
(i)	The constraint ensures that no two rows have the same value in the specified columns.	This constraint declares a field as a Primary Key which helps to uniquely identify a record.
(ii)	The UNIQUE constraint can be applied only to fields that have also been declared as NOT NULL.	The Primary Key does not allow NULL values.

22. (i) A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.
(ii) A CSV file is also known as a Flat File that can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOfficeCalc.
23. Data Visualization is the graphical representation of information and data. The objective of Data Visualization is to communicate information visually to users. For this, data visualization uses statistical graphics. Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message.
24. <opts>, <args> = getopt.getopt(argv, options, [long_options])

PART - III**25. Characteristics of an Algorithm:**

- | | |
|-------------------|--------------------|
| (i) Input | (ii) Output |
| (iii) Finiteness | (iv) Definiteness |
| (v) Effectiveness | (vi) Correctness |
| (vii) Simplicity | (viii) Unambiguous |
| (ix) Feasibility | (x) Portable |
| (xi) Independent | |

26.

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

27. Rules of local variable :

- (i) A variable with local scope can be accessed only within the function that it is created in.
(ii) When a variable is created inside the function, the variable becomes local to it.
(iii) A local variable only exists while the function is executing.
(iv) The formal parameters are also local to function.
28. sys.argv is the list of command-line arguments passed to the Python program. argv contains all the items that come along via the command-line input, it's basically an array holding the command-line arguments of the program.
main(sys.argv[1]) :
(i) Accepts the program file (Python program) and the input file (C++ file) as a list(array).
(ii) argv[0] contains the Python program which is need not to be passed because by default `_main_` contains source code reference.
(iii) argv[1] contains the name of the C++ file which is to be processed.
29. The WHERE clause is used to extract only those records that fulfill a specified condition.
Example : To display the different grades scored by male students from "student table"

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
```

```
cursor.execute("SELECT DISTINCT(Grade)
FROM student where gender='M'")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT:

```
('B,')
('A,')
('C,')
('D,')
```

30. (i) Data Visualization help users to analyze and interpret the data easily.
(ii) It makes complex data understandable and usable.
(iii) Various Charts in Data Visualization helps to show relationship in the data for one or more variables.
31. (i) An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic.
(ii) Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.
(iii) Python supports the following Arithmetic operators.

Operator – Operation	Examples	Result
Assume a=100 and b=10. Evaluate the following expressions		
+ (Addition)	>>> a + b	110
- (Subtraction)	>>> a - b	90
* (Multiplication)	>>> a * b	1000
/ (Division)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a //30 (Integer Division)	3

32. (i) The DISTINCT keyword is used along with the SELECT command to eliminate duplicate rows in the table. This helps to eliminate redundant data.
(ii) **For Example:** SELECT DISTINCT Place FROM Student; will display the following data as follows:
SELECT * FROM student;
Output:
Place
Chennai

```
Bangalore
Delhi
```

33. for i in range (1, 6):
for j in range (65, 65 + i)
a=chr(j)
print a
print

PART - IV

34. (a)
(i) **Parameters (and arguments)** : Parameters are the variables in a function definition and arguments are the values which are passed to a function definition.
(ii) **Parameter without Type** : Let us see an example of a function, definition :
(requires: b>=0)
(returns: a to the power of b)
let rec pow a b:=
if b=0 then 1
else a * pow a (b -1)
- In the above function definition variable 'b' is the parameter and the value which is passed to the variable 'b' is the argument. The precondition (**requires**) and postcondition (**returns**) of the function is given.
 - Note we have not mentioned any types: (**data types**). Some language compiler solves this type (**data type**) inference problem algorithmically, but some require the type to be mentioned.
 - In the above function definition if expression can return 1 in the then branch, shows that as per the **typing** rule the entire if expression has type **int**.
 - Since the if expression is of type '**int**', the function's return type also be '**int**'. '**b**' is compared to 0 with the equality operator, so '**b**' is also a type of '**int**'. Since '**a**' is multiplied with another expression using the * operator, '**a**' must be an int.
Parameter with Type : Now let us write the same function definition with types for some reason:
(requires: b > 0)
(returns: a to the power of b)
let rec pow (a: int) (b: int) : int :=
if b=0 then 1
else a * pow b (a-1)
 - When we write the type annotations for '**a**' and '**b**' the parentheses are mandatory. Generally we can leave out these annotations, because it's simpler to let the compiler infer them.

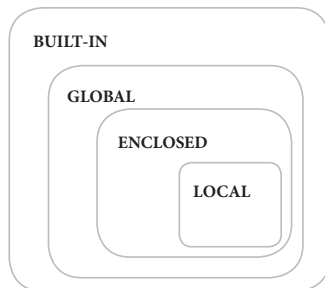
- There are times we may want to explicitly write down types. This is useful on times when you get a type error from the compiler that doesn't make sense. Explicitly annotating the types can help with debugging such an error message.

(OR)

(b) **LEGB rule :**

The **LEGB** rule is used to decide the order in which the scopes are to be searched for scope resolution. The scopes are listed below in terms of hierarchy (highest to lowest).

Local(L)	Defined inside function/class
Enclosed(E)	Defined inside enclosing functions (Nested function concept)
Global(G)	Defined at the uppermost level
Built-in(B)	Reserved names in built-in functions (modules)

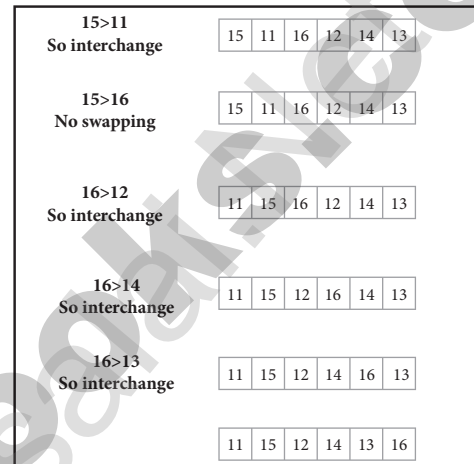
35. (a) **Bubble sort algorithm:**

- Bubble sort is a simple sorting algorithm. The algorithm starts at the beginning of the list of values stored in an array. It compares each pair of adjacent elements and swaps them if they are in the unsorted order.
- This comparison and passed to be continued until no swaps are needed, which indicates that the list of values stored in an array is sorted. The algorithm is a comparison sort, is named for the way smaller elements "bubble" to the top of the list.
- Although the algorithm is simple, it is too slow and less efficient when compared to insertion sort and other sorting methods.
- Assume list is an array of n elements. The swap function swaps the values of the given array elements.

Pseudo code :

- Start with the first element i.e., index = 0, compare the current element with the next element of the array.

- If the current element is greater than the next element of the array, swap them.
- If the current element is less than the next or right side of the element, move to the next element. Go to Step 1 and repeat until end of the index is reached.
- Let's consider an array with values {15, 11, 16, 12, 14, 13} Below, we have a pictorial representation of how bubble sort will sort the given array.



- The above pictorial example is for iteration-1. Similarly, remaining iteration can be done. The final iteration will give the sorted array. At the end of all the iterations we will get the sorted values in an array as given below :

11	12	13	14	15	16
----	----	----	----	----	----

(OR)

(b)

(i) **GROUP BY clause :**

- The **GROUP BY** clause is used with the **SELECT** statement to group the students on rows or columns having identical values or divide the table in to groups.
- For example to know the number of male students or female students of a class, the **GROUP BY** clause may be used. It is mostly used in conjunction with aggregate functions to produce summary reports from the database.
- The syntax for the **GROUP BY** clause is
- SELECT <column-names> FROM <table-name> GROUP BY <column-name> HAVING condition];**
- To apply the above command on the student table :
- SELECT Gender FROM Student GROUP BY Gender;**

- The following command will give the below given result :

Gender

M

F

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

Gender	count(*)
M	5
F	3

(ii) **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.
 - By default **ORDER BY** sorts the data in ascending order.
 - We can use the keyword **DESC** to sort the data in descending order and the keyword **ASC** to sort in ascending order.
- The **ORDER BY** clause is used as :
- SELECT <column-name>[,<column-name>,...] FROM <table-name>ORDER BY<column1>,<column2>,...ASC| DESC ;**
- For example :**
To display the students in alphabetical order of their names, the command is used as
- SELECT * FROM Student ORDER BY Name;**
The above student table is arranged as follows :

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
100	Ashish	M	17	Chennai
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai
105	Revathi	F	19	Chennai

36. (a) **Input and Output Functions :** 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.

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 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
```
- Example 2 :** `input()` without prompt string

```
>>> city=input()
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.

(ix) **Example 3 :**

```
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 :

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

(ii) 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" .....)
```

(iii) Example :

```
>>> print ("Welcome to Python Programming")
Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
11
>>> print ("The sum = ", z)
The sum = 11
>>> print ("The sum of ", x, " and ", y, " is ", z)
The sum of 5 and 6 is 11
```

(iv) The print () evaluates the expression before printing it on the monitor.

(v) 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.

(OR)

- (b) (i) Specifies label for x-axis
(ii) Specifies label for y-axis
(iii) Specifies title to the graph or assigns the plot title.
(iv) Invoke the default legend with plt
(v) Display the plot

37. (a)

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

(ii) Syntax:

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

(iii) The counter_variable mentioned in the syntax is similar to the control variable that we used in the for loop of C++ and the sequence refers to the initial, final

and increment value. Usually in Python, for loop uses the range() function in the sequence to specify the initial, final and increment values. range() generates a list of values starting from start till stop-1.

(iv) 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.

Examples for range() :

range (1,30,1) - will start the range of values from 1 and end at 29

range (2,30,2) - will start the range of values from 2 and end at 28

range (30,3,-3) - will start the range of values from 30 and end at 6

range (20) - will consider this value 20 as the end value(or upper limit) and starts the range count from 0 to 19 (remember always range() will work till stop -1 value only)

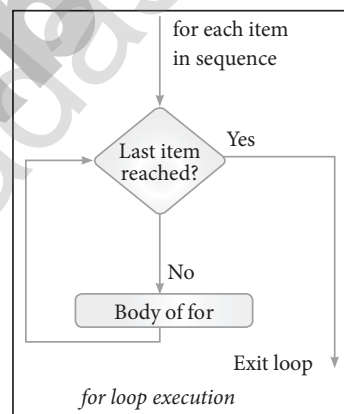
Example :

#Program to illustrate the use of for loop - to print single digit even number

```
for i in range (2,10,2):
    print (i, end=' ')
```

Output :

2 4 6 8

**(OR)**

(b)

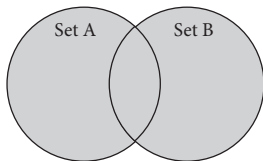
Excel	CSV
Excel is a binary file that holds information about all the worksheets in a file, including both content and formatting	CSV format is a plain text format with a series of values separated by commas.
XLS files can only be read by applications that have been especially written to read their format, and can only be written in the same way.	CSV can be opened with any text editor in Windows like notepad, MS Excel, OpenOffice, etc.

Excel is a spreadsheet that saves files into its own proprietary format viz. xls orxlsx	CSV is a format for saving tabular information into a delimited text file with extension .csv
Excel consumes more memory while importing data	Importing CSV files can be much faster, and it also consumes less memory

38. (a) A set is a mutable and an unordered collection of elements without duplicates.

Set operations: The set operation such as Union, Intersection, difference and symmetric difference.

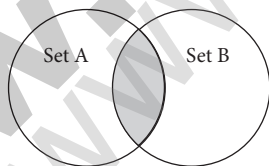
Union: It includes all elements from two or more sets



- (i) In python, the operator `|` is used to union of two sets. The function `union()` is also used to join two sets in python.
- (ii) **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'}
Intersection :
- (i) It includes the common elements in two sets



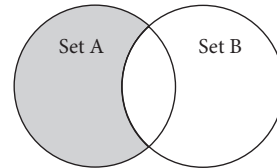
- (ii) The operator `&` is used to intersect two sets in python. The function `intersection()` is also used to intersect two sets in python.
- (iii) **Example :** Program to insect two sets using intersection operator

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

Output :
{'A', 'D'}

Difference :

- (i) It includes all elements that are in first set (say set A) but not in the second set (say set B)



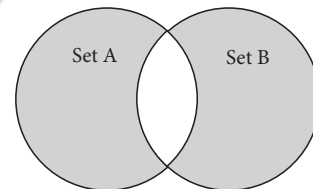
- (ii) The minus (`-`) operator is used to difference set operation in python. The function `difference()` is also used to difference operation.
- (iii) **Example :** Program to difference of two sets using minus operator

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

Output : {2, 4}

Symmetric difference :

- (i) 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.



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

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

Output :
{2, 4, 'B', 'C'}
(OR)
- (b) Python `<filename.py> -i <C++ filename without cpp extension>`
where,

Python	Keyword to execute the Python program from command-line
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

