PUBLIC EXAMINATION MAY – 2022
12TH COMPUTER SCIENCE
ANSWER KEY

1 Marks:

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 Line Feed

14. C) Boost

15. D) Relational Database System

2 Marks:

16. Syntax of if…else:

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.

17. Mapping:

The process of binding a variable name with an object is called mapping.

: = (Becomes sign) is used in programming languages to map the variable and object.

18. Searching and its types:

To search an item in a data structure is called Searching.

Types: 1. Linear Search

2. Binary search.

19. Different methods used to test Python:

1. Interactive Mode

2. Script Mode

1

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

20. Syntax of creating a tuple with n number of elements:

\# Tuple with n number elements

Tuple_Name = (E1, E2, E3……. En)

\# Elements of a tuple without parenthesis

Tuple_Name = E1, E2, E3 ….

21. Unique Constraint vs Primary key constraint:

Unique Constraint	Primary Key Constraint
This 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. It is similar to unique constraint except that only one field of a table can be set as primary key.

22. CSV File:

A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.

23. Data Visualization:

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. Syntax of getopt.getopt Method:

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

3 Marks:

25. Characteristics of an Algorithm:

Input, Output, Finiteness, Definiteness, Effectiveness, Correctness, Simplicity, Unambiguous, Feasibility, Portable, Independent.

26.

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

27. 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.
- A local variable only exists while the function is executing.
- The formal arguments are also local to function.

2

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

28. Sys.argv:

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])	Accepts the program file (Python program) and the input file (C++ file) as a list(array). argv[0] contains the Python program which is need not to be passed because by default __main__ contains source code reference and **argv[1]** contains the name of the **C++ file** which is to be processed.

29. Use of where clause:

The WHERE clause is used to extract only those records that fulfil a specified condition.

cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")

30. Uses of Data Visualization:

• Data Visualization help users to analyse and interpret the data easily.

• It makes complex data understandable and usable.

• Various Charts in Data Visualization helps to show relationship in the data for one or more variables.

31. 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.

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
/ (Divisioin)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a//30 (Integer Division)	3

32. SQL statement using DISTINCT keyword:

The **DISTINCT** keyword is used along with the **SELECT** command to eliminate duplicate rows in the table. This helps to eliminate redundant data.

SELECT DISTINCT Place FROM Student;

33. Program:

```
for i in range(1,6,1):
        ch=65
        for j in range(ch,ch+i,1):
                a=chr(j)
                print(a,end='\t ')
        print(end='\n')\
```

3

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

5 Marks:

34. A) i) Parameters:

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

ii) Pameters without type and with type:

1. 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 post condition (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, by the typing rule the entire if expression has type int. Since the if expression has 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.

2. 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.

34 B) LEGB Rule:

Scope also defines the order in which variables have to be mapped to the object in order to obtain the value.

Example:

1. x:= 'outer x variable'

2. display():

3.   x:= 'inner x variable'

4.   print x

5. display()

Output :

       outer x variable

       inner x variable

4

Above statements give different outputs because the same variable name x resides in different scopes, one inside the function display() and the other in the upper level. The value 'outer x variable' is printed when x is referenced outside the function definition. Whereas when display() gets executed, 'inner x variable' is printed which is the x value inside the function definition. From the above example, we can guess that there is a rule followed, in order to decide from which scope a variable has to be picked.

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)

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)

BUILT-IN

GLOBAL

ENCLOSED

LOCAL
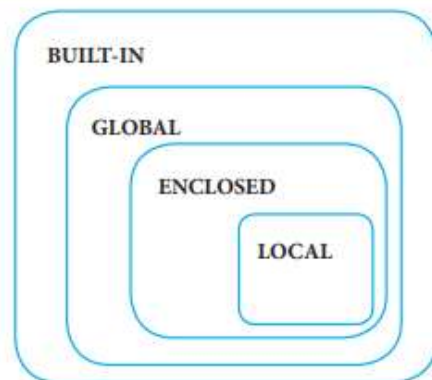
35 A) Bubble Sort algorithm:

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

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

2. If the current element is greater than the next element of the array, swap them.

5

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

3. 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 below 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 2 13 14 15 16
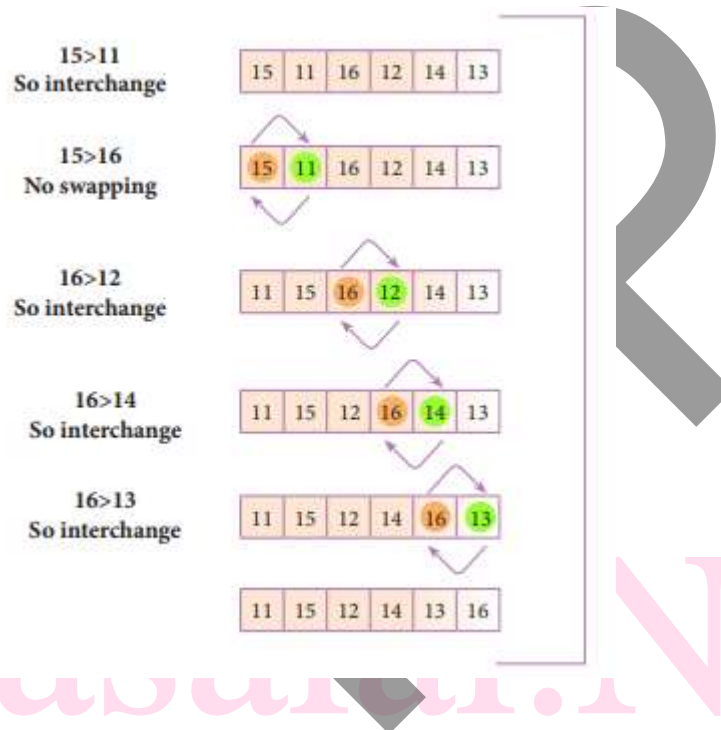
35 B) Select statement using Group by Clause:

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

6

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

The point to be noted is that only two results have been returned. This is because we only have two gender types 'Male' and 'Female'. The GROUP BY clause grouped all the 'M' students together and returned only a single row for it. It did the same with the 'F' students. For example to count the number of male and female students in the student table, the following command is given :
SELECT Gender, count(*) FROM Student GROUP BY Gender;

Gender	count(*)
M	5
F	3

Select statement using ORDER BY Clause:

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

1. By default ORDER BY sorts the data in ascending order.

2. 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;

36. A) The print() function

In Python, the print() function is used to display result on the screen.

The syntax for print() is as follows:

print ("string to be displayed as output ")

print ("String1 ", variable, "String 2", variable, "String 3" ……)

Example

>>> print ("Welcome to Python Programming")

Welcome to Python Programming
>>> print ("The sum of ", x, " and ", y, " is ", z)

The sum of 5 and 6 is 11

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.

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")

7

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.

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 :

input( ) with prompt string

>>> city=input ("Enter Your City: ")

Enter Your City: Madurai

>>> print ("I am from ", city)

I am from Madurai

36. B)

i)　　　Plt.xlabel – specifies label for X-axis
ii)　　Plt.ylabel – specifies label for Y-axis
iii)　　Plt.title – specifies title to the graph
iv)　　Plt.legend () – It invokes the default legends
v)　　　Plt.show() – it displays the chart

37. A) 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.

Syntax:
for counter_variable in sequence:
　　　statements-block 1
[else:
　　　statements-block 2]

range() generates a list of values starting from **start** till **stop-1.**

Syntax of range:

range (start,stop,[step])

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

Example:

　　　for i in range(2,10,2):
　　　　　print (i,end=' ')
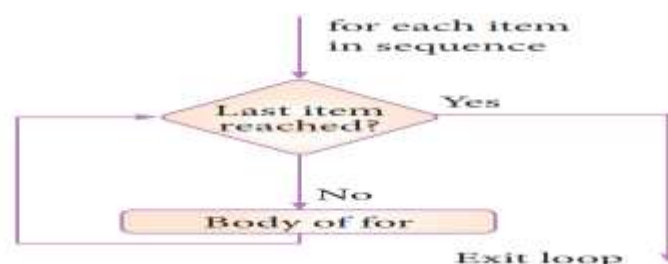　　　else:
　　　　　print ("\nEnd of the loop")

Output:
2 4 6 8
End of the loop

8

37 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 or xlsx	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) Set Operations:

(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:
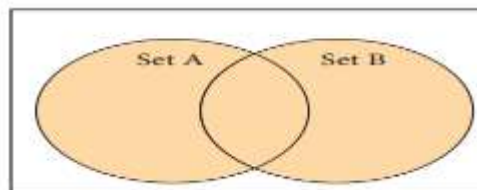set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
 U_set=set_A|set_B
print(U_set)
Output: {2, 4, 6, 8, 'A', 'D', 'C', 'B'}

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

The operator & is used to intersect two sets in python. The function intersection( ) is also used to intersect two sets in python.
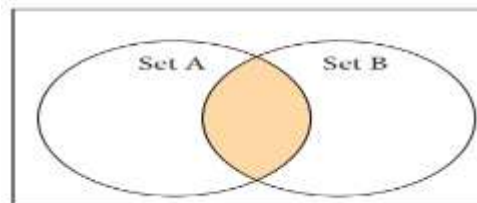
Example:
 set_A={'A', 2, 4, 'D'}
 set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
Output:
 {'A', 'D'}

iii) Difference:It includes all elements that are in first set but not in the second set

The minus (-) operator is used to difference set operation in python.
The function difference( ) is also used to difference operation.

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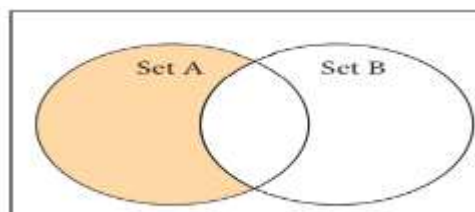set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
Output:
{2, 4}

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.

9

Prepared by: PARKUNAN T, PG Assistant in Computer Science, MHSS, Chengam. Cell: 9655 966 906.
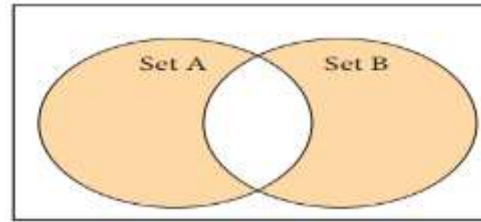
The caret (^) operator is used to symmetric difference set operation in python. The function symmetric_difference( ) is also used to do the same operation.

Example:

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

38 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.

For example type Python pycpp.py –i pali in the command prompt and press enter key.
If the compilation is successful you will get the desired output. Otherwise the error will be displayed.

Prepared by
PARKUNAN T
PG Assistant in Computer Science,
MHSS, Chengam-606 701
Cell: 9655 966 906.

10