12th STD

INSTANT SUPPLEMENTARY EXAM JUNE - 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) 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. The variables in a function definition are called as :
(a) Subroutines (b) Function
(c) Definition (d) Parameters

2. A sequence of immutable objects is called :
(a) Built in (b) List
(c) Tuple (d) Derived data

3. Containers for mapping names of variables to object is called ______
(a) Scope (b) Mapping
(c) Binding (d) Namespaces

4. The two main factors which decide the efficiency of an algorithm are ___
(a) Processor and Memory
(b) Complexity and Capacity
(c) Time and Space
(d) Data and space

5. The Shortcut key used to create new Python program is :
(a) Ctrl+C (b) Ctrl+F
(c) Ctrl+B (d) Ctrl+N

6. "elif" can be considered to be the abbreviation of :
(a) nested if (b if..else
(c) else if (d) if..elif

7. The function which is called anonymous un-named function:
(a) Lambda (b) Recursion
(c) Function (d) Define

8. ______ is used as placeholders or replacement fields which get replaced along with format( ) function
(a) { } (b) < > (c) ** (d) ^^

9. Pick odd one in connection with collection data type.
(a) List (b) Tuple
(c) Dictionary (d) Loop

10. The process of creating an object is called as:
(a) Constructor (b) Destructor
(c) Initialize (d) Instantiation

11. A table is known as _____.
(a) tuple (b) attribute
(c) relation (d) entity

12. The clause used to sort data in a database :
(a) SORT BY (b) ORDER BY
(c) GROUP BY (d) SELECT

13. The command used to skip a row in a CSV file :
(a) next ( ) (b) skip ( )
(c) omit ( ) (d) bounce ( )

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

15. The most commonly used statement in SQL is :
(a) cursor (b) select
(c) execute (d) commit

PART - II

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

16. Differentiate - Interface and Implementation.
17. What is a pair? Give an example.
18. Write short notes on Tokens.
19. List the control structures in Python.
20. What are the main advantages of function?
21. What is data consistency?
22. Write the difference between table constraint and column constraint.
23. Write notes on : (i) MAX( ) function
(ii) MIN( ) function
24. What is Set in Python?

PART - III

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

25. Write any three characteristics of modules.
26. Write a note on Asymptotic notation.
27. What are string literals?
28. Write a note on if..else Structure.

29. Write about composition in functions with an example.

30. What are the differences between List and Dictionary?

31. What is the role of DBA?

32. Differentiate Python and C++.

33. Write a Python code to check whether a given year is leap year or not.

PART - IV

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

34. (a) What is Binary search? Explain with an example.

(OR)

(b) How will you facilitate data abstraction? Explain it with suitable example.

35. (a) Explain input( ) and print( ) functions with examples.

(OR)

(b) Explain the scope of variables with an example.

36. (a) Explain about string operators in Python with suitable example.

(OR)

(b) What are the different ways to insert an element in a list? Explain with suitable example.

37. (a) Explain the characteristics of DBMS.

(OR)

(b) Explain about SQLite and the steps to be used.

38. (a) Write the different methods to read a file in Python.

(OR)

(b) Write any five features of Python.

❁❁❁❁❁

ANSWER

PART - I

1.	(d)	Parameters
2.	(c)	Tuple
3.	(d)	Namespaces
4.	(c)	Time and space
5.	(d)	Ctrl+N
6.	(c)	else if
7.	(a)	Lambda
8.	(a)	{ }
9.	(d)	Loop
10.	(d)	Instantiation
11.	(c)	relation
12.	(b)	ORDER BY
13.	(a)	next( )
14.	(a)	OS module
15.	(b)	select

PART - II

16. The difference between interface and implementation is

Interface	Implementation
Interface just defines what an object can do, but won't actually do it.	Implementation carries out the instructions defined in the interface.

17. (i) 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.

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

18. 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.

19. There are three important control structures are,

(i) Sequential (ii) Alternative or Branching

(iii) Iterative or Looping

20. Main advantages of functions are

(i) It avoids repetition and makes high degree of code reusing.

(ii) It provides better modularity for your application.

21. Data Consistency means that data values are the same at all instances of a database.

22.

Table constraint	Column constraint
Table constraint is apply to a group of one or more columns.	Column constraint apply only to individual column.

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

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

24. (i) In Python, a set is another type of collection data type. A Set is a mutable and an unordered collection of elements without duplicates.

(ii) That means the elements within a set cannot be repeated. This feature used to include membership testing and eliminating duplicate elements.

PART - III

25. The following are the desirable characteristics of a module.

(i) Modules contain instructions, processing logic, and data.

(ii) Modules can be separately compiled and stored in a library.

(iii) Modules can be included in a program.

26. Asymptotic Notations are languages that uses meaningful statements about time and space complexity. The following three asymptotic notations are mostly used to represent time complexity of algorithms:

(i) Big O : Big O is often used to describe the worst-case of an algorithm.

(ii) Big Ω : Big Omega is the reverse Big O, if Bi O is used to describe the upper bound (worst - case) of a asymptotic function, Big Omega is used to describe the lower bound (best-case).

(iii) Big Θ : When an algorithm has a complexity with lower bound = upper bound, say that an algorithm has a complexity O (n log n) and Ω (n log n), it's actually has the complexity Θ (n log n), which means the running time of that algorithm always falls in n log n in the best-case and worst-case.

27. (i) In Python, a string literal is a sequence of characters surrounded by quotes. Python supports single, double and triple quotes for a string.

(ii) A character literal is a single character surrounded by single or double quotes. The value with triple-quote "' "' is used to give multi-line string literal.

(iii) A Character literal is also considered as string literal in Python.

28. (i) The if.. else statement provides control to check the true block as well as the false block.

(ii) **Syntax:**

```
if <condition>:
        statements-block 1
else:
        statements-block 2
```

(iii) **if..else** statement thus provides two possibilities and the condition determines which BLOCK is to be executed.

29. (i) The value returned by a function may be used as an argument for another function in a nested manner.

(ii) This is called composition. For example, if we wish to take a numeric value or an expression as a input from the user, we take the input string from the user using the function input() and apply eval() function to evaluate its value.

30. (i) List is an ordered set of elements. But, a dictionary is a data structure that is used for matching one element (Key) with another (Value).

(ii) The index values can be used to access a particular element. But, in dictionary key represents index. Remember that, key may be a number of a string.

(iii) Lists are used to look up a value whereas a dictionary is used to take one value and look up another value.

31. Database Administrator or DBA is the one who manages the complete database management system. DBA takes care of the security of the DBMS, managing the license keys, managing user accounts and access etc.

32.

S.No	PYTHON	C++
(i)	Python is typically an "interpreted" language	C++ is typically a "compiled" language
(ii)	Python is a dynamic-typed language	C++ is compiled statically typed language
(iii)	Data type is not required while declaring variable	Data type is required while declaring variable
(iv)	It can act both as scripting and general purpose language	It is a general purpose language

33. **Code :**

```
n=int(input("Enter the year"))
if(y%4==0):
    print ("Leap Year")
else:
    print ("Not a Leap Year")
```

Output :

```
Enter the year    2012
Leap Year
```

PART - IV

34. **(a)** **Binary search :** Binary search also called half-interval search algorithm. It finds the position of a search element within a sorted array. The binary search algorithm can be done as divide-and-conquer search algorithm and executes in logarithmic time.

Pseudo code for Binary search :

Start with the middle element:

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

- If not, then compare the middle element with the search value,
- If the search element is greater than the number in the middle index, then select the elements to the right side of the middle index, and go to Step-1.
- If the search element is less than the number in the middle index, then select the elements to the left side of the middle index, and start with Step-1.

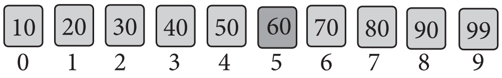
(ii) When a match is found, display success message with the index of the element matched.

(iii) If no match is found for all comparisons, then display unsuccessful message.

Binary Search Working principles :

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

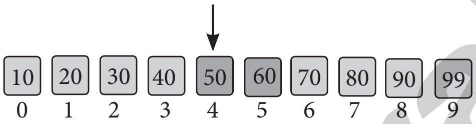
(ii) Consider the following array of elements, the array is being sorted so it enables to do the binary search algorithm. Let us assume that the search element is 60 and we need to search the location or index of search element 60 using binary search.



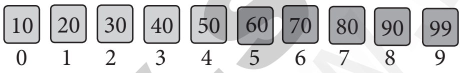
(iii) First, we find index of middle element of the array by using this formula :

mid = low + (high - low) / 2

(iv) Here it is, 0 + (9 - 0) / 2 = 4 (fractional part ignored). So, 4 is the mid value of the array.



(v) Now compare the search element with the value stored at mid value location 4. The value stored at location or index 4 is 50, which is not match with search element. As the search value 60 is greater than 50.

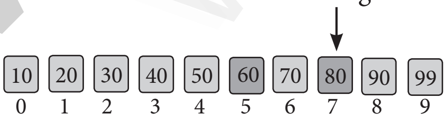


(vi) Now we change our low to mid + 1 and find the new mid value again using the formula.

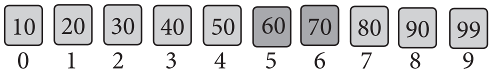
low to mid + 1

mid = low + (high - low) / 2

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



(viii) The value stored at location or index 7 is not a match with search element, rather it is more than what we are looking for. So, the search element must be in the lower part from the current mid value location

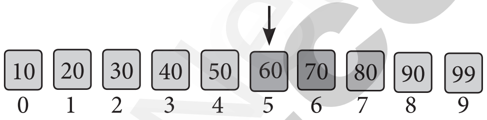


(ix) The search element still not found. Hence, we calculated the mid again by using the formula.

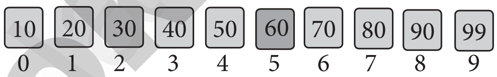
high = mid -1

mid = low + (high - low)/2

Now the mid value is 5.



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



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

(OR)

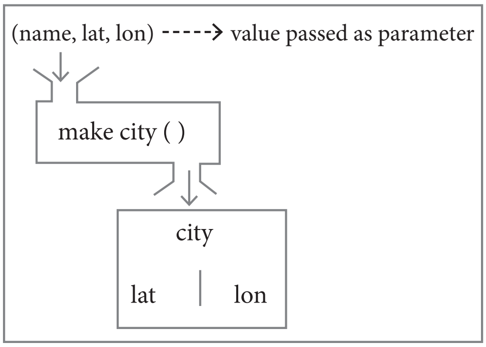
(b) Data abstraction is supported by defining an abstract data type (ADT), which is a collection of constructors and selectors. To facilitate data abstraction, you will need to create two types of functions: **Constructors and Selectors**

Constructors :

(i) Constructors are functions that build the abstract data type.

(ii) Constructors create an object, bundling together different pieces of information.

(iii) For example, say you have an abstract data type called city.



Constructor

(iv) This city object will hold the city's name, and its latitude and longitude.

(v) To create a city object, you'd use a function like **city = makecity (name, lat, lon)**.

(vi) Here makecity (name, lat, lon) is the constructor which creates the object city.

Selectors :

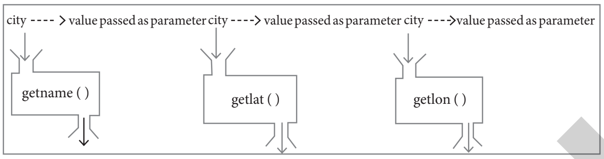
(i) Selectors are functions that retrieve information from the data type.

(ii) Selectors extract individual pieces of information from the object.

(iii) To extract the information of a city object, you would used functions like

getname(city)
getlat(city)
getlon(city)

These are the selectors because these functions extract the information of the city object.


city ----> value passed as parameter city ----> value passed as parameter city --->value passed as parameter
getname ( )
getlat ( )
getlon ( )

35. (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 :

(i) In Python, input( ) function is used to accept data as input at run time. The syntax for input() function is,

Variable = input (“prompt string”)

(ii) Where, prompt string in the syntax is a statement or message to the user, to know what input can be given.

(iii) 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.

(iv) 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.

(v) **Example 1 :** input( ) with prompt string

```
>>> city=input (“Enter Your City: ”)
 Enter Your City: Madurai
>>> print (“I am from “, city)
 I am from Madurai
```

(vi) **Example 2 :** input( ) without prompt string

```
>>> city=input()
     Madurai
>>> print ("I am from", city)
     I am from Madurai
```

(vii) 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( ).

(viii)The input ( ) accepts all data as string 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) Scope of variable refers to the part of the program, where it is accessible, i.e., area where the variables refer (use). The scope holds the current set of variables and their values. The two types of scopes - local scope and global scope.

Local Scope : A variable declared inside the function's body is known as local variable.

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.

(v) **Example:** Create a Local Variable

```
def loc():
 y=0 # local scope
 print(y)
loc()
```

Output:

```
0
```

Global Scope : A variable, with global scope can be used anywhere in the program. It can be created by defining a variable outside the scope of any function.

Rules of global Keyword :

The basic rules for global keyword in Python are:

(i) When we define a variable outside a function, it's global by default. You don't have to use global keyword.

(ii) We use global keyword to modify the value of the global variable inside a function.

(iii) Use of global keyword outside a function has no effect.

Example : Accessing global Variable From Inside a Function

```
c = 1       # global variable
def add():
                print(c)
                add()
```

Output:

```
1
```

36. (a) **String Operators :** Python provides the following operators for string operations. These operators are useful to manipulate string.

(i) **Concatenation (+) :** Joining of two or more strings is called as Concatenation. The plus (+) operator is used to concatenate strings in python.

Example :

```
>>> "welcome" + "Python"
    'welcomePython'
```

(ii) **Append (+ =) :** Adding more strings at the end of an existing string is known as append. The operator += is used to append a new string with an existing string.

Example :

```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
    Welcome to Learn Python
```

(iii) **Repeating (*) :** The multiplication operator (*) is used to display a string in multiple number of times.

Example :

```
>>> str1="Welcome "
>>> print (str1*4)
Welcome Welcome Welcome  Welcome
```

(iv) **String slicing :**

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

General format of slice operation :

```
str[start:end]
```

- Where **start** is the beginning index and **end** is the last index value of a character in the string.
- Python takes the end value less than one from the actual index specified.

Example :

slice a single character from a string

```
>>>str1="THIRUKKURAL"
>>>print (str1[0])
```

Output :

```
T
```

(v) **Stride when slicing string :**

- When the slicing operation, you can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string.

- The default value of stride is 1.
- Python takes the last value as n-1.
- You can also use negative value as stride, to prints data in reverse order.

Example :

```
>>>str1="Welcome to learn Python"
>>>print (str1[::-2])
```

Output :

```
nhy re teolW
```

(OR)

(b) Inserting elements in a list using insert():

(i) append() function in Python is used to add more elements in a list. But, it includes elements at the end of a list.

(ii) If you want to include an element at your desired position, you can use insert() function. The insert() function is used to insert an element at any position of a list.

Syntax :

List.insert (position index, element)

Example :

```
>>> MyList=[34,98,47,'Kannan', 'Gowrisankar',
                              'Lenin', 'Sreenivasan' ]
>>> print(MyList)
      [34, 98, 47, 'Kannan', 'Gowrisankar', 'Lenin',
                                      'Sreenivasan']
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
[34, 98, 47, 'Ramakrishnan', 'Kannan',
            'Gowrisankar', 'Lenin', 'Sreenivasan']
```

Output : [34, 98, 47, 'Ramakrishnan', 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']

(i) In the above example, insert( ) function inserts a new element 'Ramakrishnan' at the index value 3, ie. at the 4th position.

(ii) While inserting a new element in between the existing elements, at a particular location, the existing elements shifts one position to the right.

37. (a) Characteristics of Database Management System

Data stored into Tables	Data is never directly stored into the database. Data is stored into tables, created inside the database. DBMS also allows to have relationship between tables which makes the data more meaningful and connected.
Reduced Redundancy	In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem But DBMS follows Normalisation which divides the data in such a way that repetition is minimum.
Data Consistency	On live data, it is being continuously updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it by itself.
Support Multiple user and Concurrent Access	DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
Query Language	DBMS provides users with a simple query language, using which data can be easily fetched, inserted, deleted and updated in a database.

(OR)

(b)

(i) SQLite is a simple relational database system, which saves its data in regular data files within internal memory of the computer.

(ii) It is designed to be embedded in applications, instead of using a separate database server program such as MySQLor Oracle.

Advantages :

(i) SQLite is fast, rigorously tested, and flexible, making it easier to work. Python has a native library for SQLite.

To use SQLite,

Step 1 : import sqlite3

Step 2 : create a connection using connect () method and pass the name of the database file

Step 3 : Set the cursor object cursor = connection. cursor ()

(ii) Connecting to a database in step2 means passing the name of the database to be accessed. If the database already exists the connection will open the same. Otherwise, Python will open a new database file with the specified name.

(iii) Cursor in step 3 is a control structure used to traverse and fetch the records of the database.

(iv) Cursor has a major role in working with Python. All the commands will be executed using cursor object only.

(v) To create a table in the database, create an object and write the SQL command in it.

Example : sql_comm = "SQL statement"

(vi) For executing the command use the cursor method and pass the required sql command as a parameter. Many number of commands can be stored in the sql_comm and can be executed one after other.

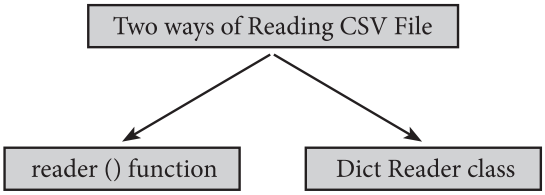
(vii) Any changes made in the values of the record should be saved by the commend "Commit" before closing the "Table connection".

38. (a) Read a CSV File Using Python :

There are two ways to read a CSV file.

(i) Use the csv module's reader function

(ii) Use the DictReader class.



CSV Module's Reader Function :

(i) You can read the contents of CSV file with the help of csv.reader() function. The reader function is designed to take each line of the file and make a list of all columns.

(ii) Then, you just choose the column you want the variable data for. Using this function one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (,).

The syntax for csv.reader() is

csv.reader(fileobject,delimiter,fmtparams)

where

(iii) **file object :** passes the path and the mode of the file

(iv) **delimiter:** an optional parameter containing the standard dilects like , | etc can be omitted.

(v) **fmtparams :** optional parameter which help to override the default values of the dialects like skipinitialspace,quoting etc. can be omitted.

Program :

```
#importing csv
import csv
#opening the csv file which is in different location
    with read mode
with open('c:\pyprg\sample1.csv', 'r',
                                newline'=') as F:
#other way to open the file is f= ('c:\\pyprg\\
                                sample1.csv', 'r')
reader = csv.reader(F)
# printing each line of the Data row by row
for row in order
    print(row)
F.close()
```

Output :

```
    ['SNO', 'NAME', 'CITY']
    ['12101', 'RAM', 'CHENNAI']
    ['12102', 'LAVANYA', 'TIRUCHY']
    ['12103', 'LAKSHMAN', 'MADURAI']
```

Reading CSV File Into A Dictionary :

(i) To read a CSV file into a dictionary can be done by using DictReader method of csv module which works similar to the reader() class but creates an object which maps data to a dictionary.

(ii) The keys are given by the fieldnames as parameter. DictReader works by reading the first line of the CSV and using each comma separated value in this line as a dictionary key. The columns in each subsequent row then behave like dictionary values and can be accessed with the appropriate key (i.e. fieldname).

(iii) If the first row of your CSV does not contain your column names, you can pass a fieldnames parameter into the DictReader's constructor to assign the dictionary keys manually.

(iv) The main difference between the csv.reader() and DictReader() is in simple terms csv.reader and csv.writer work with list/tuple, while csv.DictReader and csv.DictWriter work with dictionary. csv.DictReader and csv.DictWriter take additional argument fieldnames that are used as dictionary keys.

For Example Reading "sample8.csv" file into a dictionary

```
    import csv
    filename = 'c:\pyprg\sample8.csv'
    input_file =csv.
                    DictReader(open(filename,'r'))
for row in input_file:
    print(dict(row))        #dict() to print data
```

Output :

```
    {'ItemName ': 'Keyboard ', 'Quantity': '48'}
    {'ItemName ': 'Monitor', 'Quantity': '52'}
    {'ItemName ': 'Mouse ', 'Quantity': '20'}
```

(OR)

(b) Features of Python:

(i) Python uses Automatic Garbage Collection.

(ii) Python is a dynamically typed language.

(iii) Python runs through an interpreter.

(iv) Python code tends to be 5 to 10 times shorter than that written in C++.

(v) In Python, there is no need to declare types explicitly.

(vi) In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand.

❁❁❁❁❁