NAGALAKSHIMI MATRIC.HR. SEC. SCHOOL
SITHAYANKOTTAI – 624 708.DINDIGUL DISTRICT.
Telephone: 0451 – 2556340 Email:lakshimi.nagalakshimi@gmail.com

HIGHER SECONDARY SECOND YEAR

SECOND REVISION EXAMNATION - MARCH 2022

COMPUTER SCIENCE ANSWER KEY

Part I(Answer Key) 1-MARKS

1. (b) 3
2. (a) range()
3. (c) pass
4. (d) Recursive
5. (c) return
6. (b) Global
7. (a) 4
8. (c) string
9. (d) Hallo
10. (a) +
11. (c) len()
12. (b) [17, 23, 41, 10, 32]
13. (c) Tuples 14.
    (d) {1, 3, 6, 9}
15. (b) . (dot)

Part II (Answr Key)

16. Syntax of if...else:

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

17. Output:

c o m p u t e r

18. Main advantages of functions:

- It avoids repetition and makes high degree of code reusing.

- ❖ It provides better modularity for your application.

19. Anonymous function:

In Python, anonymous function is a function that is defined without a name. While normal functions are defined using the def keyword, in Python anonymous functions are defined using the lambda keyword. Hence, anonymous functions are also called as lambda functions.

20. String is a data type in python, used to handle array of characters. String is a sequence of characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

21. Accessing characters in a String:

Python allocate an index value for its each character. These index values are otherwise called as subscript which are used to access and manipulate the strings. The subscript can be positive or negative integer numbers.

22. Difference del and remove()

- ✔ del statement is used to delete known elements remove()function is used to delete elements of a list if its index is unknown.
- ✔ The del statement can also be used to delete entire list. The remove is used to delete a particular element.

23. Instantiation:

The process of creating object is called as "Class Instantiation".

Syntax:

Object_name = class_name( )

24. Purpose of Destructor:

- ☐ Destructor is also a special method gets executed automatically when an object exit from the scope.
- ☐ In Python, del__ ( ) method is used as destructor.

General format:

```
def del (self):

                    <statements>
```

Part –III

25. Syntax:

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

26. i=1

```
while (i<=4):
        for j in range (1,i): print
                (j,end='\t')
        print (end='\n')
```

```
i+=1
```

27.

- ♦ Function blocks begin with the keyword "def" followed by function name and parenthesis ().
- ♦ Any input parameters should be placed within these parentheses.
- ♦ The code block always comes after a colon (:) and is indented.
- ♦ The statement "return [expression]" exits a function, and it is optional.
- ♦ A "return" with no arguments is the same as return None.

28. Advantages of Tuples over a list:

- ✔ The elements of a list are changeable (mutable) whereas the elements of a tuple are unchangeable (immutable), this is the key difference between tuples and list.
- ✔ The elements of a list are enclosed within square brackets. But, the elements of a tuple are enclosed by parenthesis.
- ✔ Iterating tuples is faster than list.

29. [1, 2, 4, 8, 16]

30. Accessing elements using for loop:

In Python, the for loop is used to access all the elements in a list one by one. This is just like the for keyword in other programming language such as C++.

Syntax:

```
for index_var in list: print
        (index_var)
```

Here, index_var represents the index value of each element in the list.

31. copy() - Returns a copy of the list. List.copy( )

index() - Returns the index value of the first recurring element.

List.index(element)

count ( ) - Returns the number of similar elements present in the last.

List.count(value)

32. Output:

Total Marks = 207 Average Marks = 69.0

33. Public and Private Data Members:

- The variables which are defined inside the class is public by default. These variables can be accessed anywhere in the program using dot operator.
- A variable prefixed with double underscore becomes private in nature. These variables can be accessed only within the class.

Part -IV

34. (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:# optional block
        statements-block 2]
```

- The counter_variable is the control variable.
- The sequence refers to the initial, final and increment value.
- 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.
- 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. 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

34. (b)

```
x=int(input("Enter the number:")) if
x==0:
    print("The given number is Zero") elif
x>0:
    print("The given number is Positive")
else:
    print("The given number is Negative")
```

35. (a) Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
We can say that scope holds the current set of variables and their values.
There are two types of scopes - local scope and global scope.

Local Scope:

A variable declared inside the function's body or in the local scope is known as local variable.

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.

Example:

```
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/block.

The basic rules for global keyword in Python are:

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

We use global keyword to read and write a global variable inside a function.

Use of global keyword outside a function has no effect Without using the global keyword we cannot modify the global variable inside the function but we can only access the global variable.

Example:

```
x = 0 # global variable
def add():
    global x
    x = x + 5     # increment by 2
    print ("Inside add() function x value is :", x) add()
    print ("In main x value is :", x)
```

Output:

```
Inside add() function x value is : 5 In
main x value is : 5
```

35. (b) **The return Statement:**

The return statement causes your function to exit and returns a value to its caller. The point of functions in general is to take inputs and return something.

The return statement is used when a function is ready to return a value to its caller. So, only one return statement is executed at run time even though the function contains multiple return statements. Any number of 'return' statements are allowed in a function definition but only one of them is executed at run time.

Syntax of return

return [expression list]

This statement can contain expression which gets evaluated and the value is returned.

If there is no expression in the statement or the return statement itself is not present inside a function, then the function will return the None object.

Example:

```
# return statment
```

```
def usr_abs (n):
    if n>=0:
        return n
    else:
        return –n
# Now invoking the function x=int
(input("Enter a number :") print
(usr_abs (x))
```

Output 1:

Enter a number : 25 25

Output 2:

Enter a number : -25 25

36. (a) **STRING OPERATORS**

Python provides the following string operators to manipulate string.

(i) Concatenation (+)

Joining of two or more strings using plus (+) operator is called as Concatenation.

Example

```
>>> "welcome" + "Python"
```

Output: 'welcomePython'

(ii) Append (+ =)

Adding more strings at the end of an existing string using operator += is known as append.

Example:

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

Output: 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)
```

Output: 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 [ ] slicing operator and index values.

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[10:16])
>>> print(str1[::-2])
```

Output: learn

nhy re teolW

36. (b) **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.

For example, if you want to slice first 4 characters from a string, you have to specify it as 0 to 5. Because, python consider only the end value as n-1.

Example I : slice a single character from a string

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

Example II : slice a substring from index 0 to 4

```
>>> print (str1[0:5]) THIRU
```

Example III : slice a substring using index 0 to 4 but without specifying the beginning index.

```
>>> print (str1[:5]) THIRU
```

Example IV : slice a substring using index 0 to 4 but without specifying the end index.

```
>>> print (str1[6:]) KURAL
```

37. (a)

Inserting elements in a list using insert():

The insert ( ) function helps you to include an element at your desired position.

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' ]
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
```

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

In the above example, insert( ) function inserts a new element ‘Ramakrishnan’ at the index value 3, ie. at the 4^{th} position.

While inserting a new element, the existing elements shifts one

position to the right.

Adding more elements in a list using append():

The append( ) function is used to add a single element in a list. But, it includes elements at the end of a list.

Syntax:

List.append (element to be added)

Example:

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
```

Output: [34, 45, 48, 90]

Adding more elements in a list using extend():

The extend( ) function is used to add more than one element to an existing list.

In extend( ) function, multiple elements should be specified within square bracket as arguments of the function.

Syntax:

List.extend ([elements to be added]) Example:

```
>>> Mylist=[34, 45, 48]
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
```

Output: [34, 45, 48, 90, 71, 32, 29]

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}

(i) 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.

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. (a) **Defining classes**

In Python, a class is defined by using the keyword class. Every class has a unique name followed by a colon (:).

Syntax:

```
class class_name:
statement_1
statement_2
.....
.....
statement_n
```

Where, statement in a class definition may be a variable declaration, decision control, loop or even a function definition. Variables defined inside a class are called as “Class Variable” and functions are called as “Methods”. Class variable and methods are together known as members of the class. The class members should be accessed through objects or instance of class. A class can be defined anywhere in a Python program.

Example: Program to define a class

```
class Sample:
    x, y = 10, 20 # class variables
```

In the above code, name of the class is Sample and it has two variables x and y having the initial value 10 and 20 respectively. To access the values defined inside the class, you need an object or instance of the class.

38. (b) **Output:**

```
The object value is =  15
The count of object created = 1 The
object value is = 35
The count of object created = 2 The
object value is = 45
The count of object created = 3
```

Prepared By
K.CHELLAPANDI M.Sc.,B.Ed.,

Computer Instructor
Nagalakshimi Matric.Hr.Sec.School
Sithayankottai -624708 Dindigul District