NADAR HR.SEC.SCHOOL, RAJAPALAYAM.

XII - COMPUTER SCIENCE – ENGLISH MEDIUM

FIRST MID ANSWER KEY 2024

1.	Arguments	6.	pass
2.	Pair	7.	Lambda
3.	Namespaces	8.	0.125
4.	Big O	9.	12
5.	Guido van Rossum	10.	Program

11. Define Function with respect to Programming language.

A function is a unit of code that is often defined within a greater code structure. Specifically, a function contains a set of code that works on many kinds of inputs, like variants, expressions and produces a concrete output.

12. What is a Pair? Give an example.

- Any way of bundling two values together into one can be considered as a pair.
- Pair is a compound structure which is made up of list or Tuple.

Example: lst := [5, 10, 20]

13. How Python represents the private and protected Access specifiers?

- Python prescribes a convention of prefixing the name of the variable or method with single or double underscore to emulate the behaviour of protected and private access specifiers.
- All members in a Python class are public by default.

14. What is memoization?

Memoization is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls and returning the cached result when the same inputs occur again.

15. Write a note on comments in Python?

In Python, comments begin with hash symbol (#). The lines that begins with # are considered as comments and ignored by the Python interpreter. Comments may be single line or no multi-lines.

Example:

```
# It is Single line Comment
```

16. Define control structure.

A program statement that causes a jump of control from one part of the program to another is called control structure or control statement.

17. Write is the syntax of if .. else statement.

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

18. Differentiate Concrete data type and abstract datatype.

Concrete data type	Abstract datatype
Concrete data types or structures (CDT's) are direct implementations of a relatively simple concept.	Abstract Data Types (ADT's) offer a high level view (and use) of a concept independent of its implementation.

19. Built-in scope:

- The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.
- Any variable or module which is defined in the library functions of a programming language has Built-in or module scope.

Example:

```
Built-in/module scope →   library files
a := 10               →   global scope
Disp():
    b := 7            →   enclosed scope
    Disp1()
```

20. What are the factors that influence time and space complexity.

The efficiency of an algorithm depends on how efficiently it uses time and memory space. The time efficiency of an algorithm is measured by different factors.

- ✓ Speed of the machine
- ✓ Compiler and other system Software tools
- ✓ Operating System
- ✓ Programming language used
- ✓ Volume of data required

21. Explain Ternary operator with examples.

- 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.

Syntax:

Variable_Name = [on_true] if [Test expression] else [on_false]

Example:

```
min = 50 if 49<50 else 70
```

22. Syntax:

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

23. How recursive function works?

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

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

```
n=int(input("Enter a Year : "))
if n%4 == 0:
    print(n," is a Leap Year")
else:
    print(n," is not a Leap Year")
```

25. Explain with an example interface and implementation.

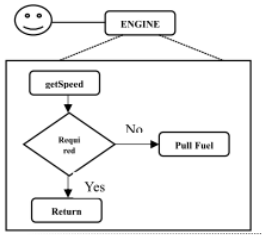
Interface:

An interface is a set of action that an object can do. Interface just defines what an object can do, but won't actually do it.

Implementation:

Implementation carries out the instructions defined in the interface.

For example, let's take the example of increasing a car's speed.



- The person who drives the car doesn't care about the internal working.
- To increase the speed of the car he just presses the accelerator to get the desired behaviour. Here the accelerator is the interface between the driver (the calling / invoking object) and the engine (the called object).
- Internally, the engine of the car is doing all the things. It's where fuel, air, pressure, and electricity come together to create the power to move the vehicle.
- All of these actions are separated from the driver, who just wants to go faster. Thus, we separate interface from implementation.

25. How will you facilitate data abstraction. Explain it with suitable example.

To facilitate data abstraction, you will need to create constructors and selectors.

- Constructors are functions that build the abstract data type.
- Selectors are functions that retrieve information from the data type.

for example

Let's take an abstract datatype called city. This city object will hold the city's name, and its latitude and longitude.

city = makecity (name, lat, lon)

- ✓ Here the function **makecity (name, lat, lon)** is the constructor. When it creates an object city, the values name, lat and lon are sent as parameters.
- ✓ **getname(city), getlat(city) and getlon(city)** are selector functions that obtain information from the object city.

26. **Write any five benefits in using modular programming.**

- Less code to be written.
- Programs can be designed more easily because a small team deals with only a small part of the entire code.
- Code is short, simple and easy to understand.
- Errors can easily be identified, as they are localized to a subroutine or function.
- The same code can be used in many applications.
- Modular programming allows many programmers to collaborate on the same application.
- The scoping of variables can easily be controlled.

26. **Explain the concept of Dynamic programming with suitable example.**

- Dynamic programming algorithm can be used when the solution to a problem can be viewed as the result of a sequence of decisions.
- Dynamic programming approach is similar to divide and conquer. The given problem is divided into smaller and yet smaller possible sub-problems.
- It is used whenever problems can be divided into similar sub-problems. so that their results can be re-used to complete the process.
- For every inner sub problem, dynamic algorithm will try to check the results of the previously solved sub-problems. The solutions of overlapped sub-problems are combined in order to get the better solution.

Steps to do Dynamic programming

- The given problem will be divided into smaller overlapping sub-problems.
- An optimum solution for the given problem can be achieved by using result of smaller sub-problem.
- Dynamic algorithms uses Memoization.

Fibonacci Series – An example

Fibonacci series generates the subsequent number by adding two previous numbers.

Fibonacci Iterative Algorithm with Dynamic programming approach

Step - 1: Print the initial values of Fibonacci f0 and f1

Step - 2: Calculate fib = f0 + f1

Step - 3: Print the value of fib

Step - 4: Assign f0 = f1, f1 = fib

Step - 5: Goto **Step - 2** and repeat until the specified number of terms generated.

If the number of terms is 10 then the output of the Fibonacci series is :

0 1 1 2 3 5 8 13 21 34 55

27. **Explain input() and print() functions with examples.**

input( ) function:

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

The syntax is

Variable = input ("prompt string")

- ✓ Where, prompt string is a statement or message to the user, to know what input can be given.

Example:

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

- ✓ input() accepts all data as string or characters but not as numbers. The int( ) function is used to convert string data as integer data explicitly.

Example:

```
x = int(input( "Enter a number:" ))
```

print() function:

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

The syntax is

```
print( "String" )
print( variable )
print( "string", variable )
print( "string1", varl, "string2"', var2)
```

Example:

```
(1) >>>print("Welcome")
        Welcome
(4) >>>x = 2
    >>>y = 3
    >>>print( " The sum is ", x+y )
        The sum is 5
```

- ✓ The print ( ) evaluates the expression before printing it on the monitor.
- ✓ Comma (,) is used as a separator in print ( ) to print more than one item.

27. **Write a Python code to find the L.C.M. of two numbers.**

Program:

```
a = int(input("Enter First Number :"))
b = int(input("Enter Second Number :"))
if a>b:
        min = a
else:
        min = b
while 1:
        if min % a ==0 and min % b ==0:
                print (min)
                break
        min = min + 1
```

Output:

```
Enter First Number :5
Enter Second Number :3
15
```