NON PROGRAMMING QUESTIONS:-
MAY 2023-(3 marks)

1-What is the output of the following Python code. Justify your answer.

```
>>>x = 'abcd'
>>>for i in range(len(x)):
>>>     print(i)
```

ANS:

```
        0
        1
        2
        3
```

2-Write the syntax and semantics of the multiway-if statement.

ANS:

Syntax:-

```
        if condition1:
                statement
        elif condition2:
                statement
        elif condition3:
                statement
        ...
        ...
        else:
                statement_block
```

Semantics:

The multiway-if statement allows a program to choose one path of execution from multiple alternatives based on evaluated conditions.

condition checking and selection of statement block

->The conditions (condition1, condition2, etc.) are evaluated in sequence, starting from the top.

->Each condition is a boolean expression that evaluates to true or false

->If a condition evaluates to true, its corresponding statement_block is executed, and the rest of the conditions are skipped.

->Only the first true condition's block is executed, even if other conditions might also be true.

->If none of the conditions evaluate to true, the else block (if present) is executed.

->The else block serves as the default case and is optional.

JUNE 2022-(3 marks)

1-What is the output of the following print statement in Python?

(a) print (9//2) (b) print (9/2)

ANS:

(a) 4

(b) 4.5

--(8 marks)

2-Mention the different types of loop and control statements allowed in Python and explain each type with suitable examples.
ANS:
->Loops in Python
Python supports two types of loops:
1-for Loop
2-while Loop

1-for loop:

-The for loop is used to iterate over a sequence (like a list, tuple, string, or range).

-syntax:

```
>>>for variable in sequence:
>>>    Body of the loop
```

-EXAMPLE:

```
>>>for num in range(1, 6):
>>>     print(i,end=" ")
OUTPUT:
        1 2 3 4 5
```

the loop print the numbers from 1 to 5

2-while loop:

-The while loop continues to execute as long as the condition evaluates to True.

-syntax:

```
>>> while <condition>:
>>>     loop_body
```

-example:

```
>>> count = 1
>>> while count <= 5:
>>>             print(f"Count: {count}",end=" ")
>>>             count += 1
        OUTPUT:
                1 2 3 4 5
```

loop execute untle count iss lessthan or equal to 5

->Control Statements in Python
Python provides the following control statements to alter the flow of loops:

1-break Statement
2-continue Statement
3-pass Statement

1-break Statement:

-The break statement is used to exit a loop prematurely when a specific condition is met.

-EXAMPLE:

```
>>> for num in range(1, 10):
>>>     if num == 5:
>>>             break
>>>     print(num,end=" ")
```

in this when num become 5 it break from the loop or exit

from the loop,

```
                        OUTPUT:
                                1 2 3 4
```

2-continue Statement:

-The continue statement skips the rest of the code inside the loop for the current iteration and proceeds to the next iteration.

-EXAMPLE:

```
                        >>> for num in range(1, 6):
                        >>>     if num == 3:
                        >>>             continue
                        >>>     print(f"Number={num})

                        OUTPUT:
                                Number: 1
                                Number: 2
                                Number: 4
                                Number: 5
```

Here when num is 3 then then the break from the loop and it will continue iteration untile the given range,thats why the print statement doesnt execute

3-pass Statement:

-The pass statement is used as a placeholder. It does nothing and is often used when the code is not yet implemented.

-EXAMPLE:

```
                        >>> for num in range(1, 4):
                        >>>     if num == 2:
                        >>>             pass
                        >>>     print(f"Number: {num}")

                        OUTPUT:
                                Number: 1
                                Number: 2
                                Number: 3
```

MAY 2024 (3 marks)

1-Let the variable x be "dog" and the variable y be "cat". Write the values returned
by the following operations: a) x*4 + ' ' + 4*y b) x*len(x+y)

ANS:

a) "dogdogdogdog catcatcatcat"

b) "dogdogdogdogdogdog"

2-What will be the output if the following code fragments are executed?

```
        >>>for j in range(2,10,4):
        >>>     print(j)
```

ANS:

```
        2
```

JANUARY 2024 -(3 marks)

1-Explain type conversion with example.

ANS:

Type conversion refers to converting one data type into another. It can be classified into two types:

->Implicit Type Conversion (Type Coercion)
->Explicit Type Conversion (Type Casting)

1-Implicit Type Conversion (Type Coercion):

>In implicit conversion, Python automatically converts one data type to another during an operation without user intervention.
>This happens when no data loss or error occurs during the conversion.
>EX:

```
>>>num_int = 10
>>>num_float = 2.5
>>>result = num_int + num_float
>>>print("Result:", result)
>>>print("Type of result:", type(result))
```

>The integer num_int is automatically converted to a float during addition.
>The result is a float because Python ensures data precision.

2-Explicit Type Conversion (Type Casting):

>This requires the programmer to explicitly specify the data type for conversion using functions like int(), float(), or str().
>EXAMPLE:

```
>>>a = 10.75
>>>b = int(a)
>>>print(b)
>>>c = 100
>>>d = str(c)
>>>print(d)
>>>print(type(d))
```

> Here a is of type float,on using the function int(),type is converted into int
>c of type int,on using str(),type is changed into string

JUNE 2023 -(3 marks)

1-Jack says that he will not bother with analysis and design but proceed directly to

coding his programs. Why is that not a good idea?

ANS:
Skipping analysis and design to proceed directly to coding can lead to several problems:
> Without analysis, Jack might not fully understand the problem, leading to incorrect or incomplete solutions.
>Without design, the program structure may be flawed, resulting in bugs and inefficiencies that are harder to fix later.
>Coding without a plan may lead to rework or scrapping parts of the program if they don't align with the requirements or goals

2-Write the output of the following python statements :
i) round(12.57) ii) 5//2 iii) int(6.5)

ANS:
i)13 ii)2 iii)6