

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

Common Half Yearly Examination, 2022 – 2023

12th – Computer Science

1	C) Definition
2	A) Scope
3	B) Average Case
4	C) Comma (,)
5	B) True
6	A) I is correct and II is wrong
7	A) { }
8	D) :
9	C) Classes and Objects
10	C) Edger Frank Codd
11	A) DROP
12	D) Carriage Return and Line Feed
13	C) args variable
14	C) Commit
15	C) Install PIP

16	<ul style="list-style-type: none"> ➤ 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. ➤ The definition of ADT only mentions what operations are to be performed but not how these Operations will be implemented. ➤ ADT does not specify how data will be organized in memory and what algorithms will be used for implementing the operations
----	---

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

17	<ul style="list-style-type: none"> ➤ The process of binding a variable name with an object is called mapping. ➤ = (equal to sign) is used in programming languages to map the variable and object. 				
18	<ul style="list-style-type: none"> ➤ Literal is a raw data given in a variable or constant. In Python, there are various types of literals. 1) Numeric 2) String 3) Boolean 				
19	<ul style="list-style-type: none"> ➤ Python will not allow deleting a particular character in a string. ➤ Whereas you can remove entire string variable using del command. 				
20	<ul style="list-style-type: none"> ➤ 4 				
21	<ul style="list-style-type: none"> ➤ Data Consistency means that data values are the same at all instances of a database 				
22	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 50%;">Table Constraint</th> <th style="width: 50%;">Column Constraint</th> </tr> </thead> <tbody> <tr> <td>Table constraint apply to a group of one or more columns</td> <td>Column constraint apply only to individual column.</td> </tr> </tbody> </table>	Table Constraint	Column Constraint	Table constraint apply to a group of one or more columns	Column constraint apply only to individual column.
Table Constraint	Column Constraint				
Table constraint apply to a group of one or more columns	Column constraint apply only to individual column.				
23	<ul style="list-style-type: none"> ➤ connection method used to connect database ➤ Example: <code>connection = sqlite3.connect ("Academy.db")</code> 				
24	<ul style="list-style-type: none"> ➤ Line plot ➤ Scatter plot ➤ Histogram ➤ Box plot ➤ Bar chart and ➤ Pie chart 				
25	<table style="width: 100%; border: none;"> <thead> <tr> <th style="width: 50%; text-align: center;">Pure Function</th> <th style="width: 50%; text-align: center;">Impure Function</th> </tr> </thead> <tbody> <tr> <td style="border: none; vertical-align: top;"> <p>The return value of the pure functions solely depends on its arguments passed. Hence, if you call the pure functions with the same set of arguments, you will always get the same return values.</p> <p>They do not have any side effects.</p> <p>They do not modify the arguments which are passed to them</p> </td> <td style="border: none; vertical-align: top;"> <p>The return value of the impure functions does not solely depend on its arguments passed. Hence, if you call the impure functions with the same set of arguments, you might get the different return values For example, <code>random()</code>, <code>Date()</code>.</p> <p>They may modify the arguments which are passed to them</p> </td> </tr> </tbody> </table>	Pure Function	Impure Function	<p>The return value of the pure functions solely depends on its arguments passed. Hence, if you call the pure functions with the same set of arguments, you will always get the same return values.</p> <p>They do not have any side effects.</p> <p>They do not modify the arguments which are passed to them</p>	<p>The return value of the impure functions does not solely depend on its arguments passed. Hence, if you call the impure functions with the same set of arguments, you might get the different return values For example, <code>random()</code>, <code>Date()</code>.</p> <p>They may modify the arguments which are passed to them</p>
Pure Function	Impure Function				
<p>The return value of the pure functions solely depends on its arguments passed. Hence, if you call the pure functions with the same set of arguments, you will always get the same return values.</p> <p>They do not have any side effects.</p> <p>They do not modify the arguments which are passed to them</p>	<p>The return value of the impure functions does not solely depend on its arguments passed. Hence, if you call the impure functions with the same set of arguments, you might get the different return values For example, <code>random()</code>, <code>Date()</code>.</p> <p>They may modify the arguments which are passed to them</p>				

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

26	<ul style="list-style-type: none"> ➤ Input ➤ Output ➤ Finiteness ➤ Definiteness ➤ Effectiveness ➤ Correctness ➤ Simplicity ➤ Unambiguous ➤ Feasibility ➤ Portable ➤ Independent 				
27	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; width: 50%;">break</th> <th style="text-align: center; width: 50%;">Continue</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> ➤ The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop ➤ If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop. </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> ➤ Continue statement unlike the break statement is used to skip the remaining part of a loop and start with next iteration. </td> </tr> </tbody> </table>	break	Continue	<ul style="list-style-type: none"> ➤ The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop ➤ If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop. 	<ul style="list-style-type: none"> ➤ Continue statement unlike the break statement is used to skip the remaining part of a loop and start with next iteration.
break	Continue				
<ul style="list-style-type: none"> ➤ The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop ➤ If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop. 	<ul style="list-style-type: none"> ➤ Continue statement unlike the break statement is used to skip the remaining part of a loop and start with next iteration. 				
28	<ul style="list-style-type: none"> ➤ 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 formate arguments are also local to function. 				
29	Good Morning Bindhu Madhavan				
30	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; width: 50%;">write mode</th> <th style="text-align: center; width: 50%;">append mode</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> ➤ Open a file for writing. Creates a new truncates the file if it exists. </td> <td style="vertical-align: top;"> Open for appending at the end of the file wi file if it does not exist. </td> </tr> </tbody> </table>	write mode	append mode	<ul style="list-style-type: none"> ➤ Open a file for writing. Creates a new truncates the file if it exists. 	Open for appending at the end of the file wi file if it does not exist.
write mode	append mode				
<ul style="list-style-type: none"> ➤ Open a file for writing. Creates a new truncates the file if it exists. 	Open for appending at the end of the file wi file if it does not exist.				
31	<ul style="list-style-type: none"> ➤ To automate certain tasks in a program ➤ Extracting information from a data set ➤ Less code intensive as compared to traditional programming language ➤ can bring new functions to applications and glue complex systems together 				

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

32	<ul style="list-style-type: none"> ➤ SELECT * FROM Employee ORDER BY Eno DESC;
33	<ul style="list-style-type: none"> ➤ Data Visualization help users to analyze 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.
34	<p><u>Parameter without type</u></p> <p><i>(requires: $b \geq 0$)</i> <i>(returns: a to the power of b)</i> <i>let rec pow a b:=</i> <i>if $b=0$ then 1</i> <i> else $a * \text{pow } a (b-1)$</i></p> <ul style="list-style-type: none"> ➤ 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 (<i>requires</i>) and postcondition (<i>returns</i>) of the function is given. ➤ we have not mentioned any types: (<i>data types</i>). <p><u>Parameter with Type</u></p> <p><i>(requires: $b > 0$)</i> <i>(returns: a to the power of b)</i> <i>let rec pow (a: int) (b: int) : int :=</i> <i> if $b=0$ then 1</i> <i> else $a * \text{pow } b (a-1)$</i></p> <ul style="list-style-type: none"> ➤ In the above function definition variable 'b' is the parameter and the value which is passed to the variable 'b' is the argument. ➤ All the arguments and return value also mentioned with type of data(int). ➤ When we write the type annotations for 'a' and 'b' the parentheses are mandatory. This is called parameter with type.
Or	<p><u>Binary search</u> Page No : 37 and 38</p>

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

- **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 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 of the loop and the sequence refers to the initial, final and increment value.
- In Python, **for** loop uses the range() function in the sequence to specify the initial, final and
- increment values.

35

a **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

Scope of variable

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- There are two types of scopes - **local scope** and **global scope**.

35

b

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.

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

- A local variable only exists while the function is executing.
- The formate 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.

Rules of global Keyword

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

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
```

- The python supports the set operations such as Union, Intersection, difference and Symmetric difference.

(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 :

```
a={2,4,6,8}
b={'A', 'B', 'C', 'D'}
print(A|B )
```

Output:

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

36
a

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

(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:

```
a={'A', 2, 4, 'D'}
b={'A', 'B', 'C', 'D'}
print(a & b)
```

Output:

```
{'A', 'D'}
```

(iii) Difference:

- It includes all elements that are in first set (say set A) but not in the second set (say set B)
- The minus (-) operator is used to difference set operation in python.
- The function **difference()** is also used to difference operation

Example :

```
a={'A', 2, 4, 'D'}
b={'A', 'B', 'C', 'D'}
print(a - 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.
- 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 :

```
a={'A', 2, 4, 'D'}
b={'A', 'B', 'C', 'D'}
print(a ^ b)
```

Output:

```
{2, 4, 'B', 'C'}
```

Types of Relationships

1. One-to-One Relationship
2. One-to-Many Relationship
3. Many-to-One Relationship
4. Many-to-Many Relationship

36
b

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

1. One-to-One Relationship

In One-to-One Relationship, one entity is related with only one other entity. One row in a table is linked with only one row in another table and vice versa.

For example: A student can have only one exam number

2. One-to-Many Relationship

In One-to-Many relationship, one entity is related to many other entities.

One row in a table A is linked to many rows in a table B, but one row in a table B is linked to only one row in table A.

For example: One Department has many staff members

3. Many-to-One Relationship

In Many-to-One Relationship, many entities can be related with only one in the other entity.

For example: A number of staff members working in one Department.

Multiple rows in staff members table is related with only one row in Department table.

4. Many-to-Many Relationship

A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table

Example 1: Customers and Product

Customers can purchase various products and Products can be purchased by many customers

Example 2: Students and Courses

A student can register for many Courses and a Course may include many students

37
a

- i) `SELECT * FROM employee ORDER BY PAY DESC;`
- ii) `SELECT * FROM employee WHERE ALLOWANCE BETWEEN 5000 AND 7000`
- iii) `DELETE FROM employee WHERE DESIG ='Mechanic'`
- iv) `INSERT INTO employee(EMP CODE, NAME,DESIG,PAY,ALLOWANCE) VALUES (M1006,SHAN, Mechanic,25000,5000);`
- v) `SELECT * FROM employee WHERE DESIG ='Operator'`

St. Paul's Mat. Hr. Sec. School, Block – 4, Neyveli

37 b	Mode	Description
	'r'	Open a file for reading. (default)
	'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
	'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
	'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
	't'	Open in text mode. (default)
	'b'	Open in binary mode.
	'+'	Open a file for updating (reading and writing)
38 a	<p><u>Features of Python</u></p> <ul style="list-style-type: none"> ➤ Python uses Automatic Garbage Collection whereas C++ does not. ➤ Python is a dynamically typed language. ➤ Python runs through an interpreter ➤ Python code tends to be 5 to 10 times shorter ➤ Python, there is no need to declare types explicitly where ➤ In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand. 	
38 b	<p>Home Button</p> <p>The Home Button will help once you have begun navigating your chart. If you ever want to return back to the original view, you can click on this.</p> <p>Forward/Back buttons</p> <p>These buttons can be used like the Forward and Back buttons in your browser. You can click these to move back to the previous point you were at, or forward again.</p> <p>Pan Axis</p> <p>This cross-looking button allows you to click it, and then click and drag your graph around.</p> <p>Zoom</p> <p>The Zoom button lets you click on it, then click and drag a square that you would like to zoom into specifically. Zooming in will require a left click and drag. You can alternatively zoom out with a right click and drag.</p> <p>Configure Subplots</p> <p>This button allows you to configure various spacing options with your figure and plot.</p> <p>Save Figure</p> <p>This button will allow you to save your figure in various forms.</p>	