Padasalai's

Telegram Groups!

(தலைப்பிற்கு கீழே உள்ள லிங்கை கிளிக் செய்து குழுவில் இணையவும்!)

- Padasalai's NEWS - Group

https://t.me/joinchat/NIfCqVRBNj9hhV4wu6_NqA

- Padasalai's Channel - Group

https://t.me/padasalaichannel

- Lesson Plan - Group

https://t.me/joinchat/NIfCqVWwo5iL-21gpzrXLw

- 12th Standard - Group

https://t.me/Padasalai_12th

- 11th Standard - Group

https://t.me/Padasalai_11th

- 10th Standard - Group

https://t.me/Padasalai_10th

- 9th Standard - Group

https://t.me/Padasalai_9th

- 6th to 8th Standard - Group

https://t.me/Padasalai_6to8

- 1st to 5th Standard - Group

https://t.me/Padasalai_1to5

- TET - Group

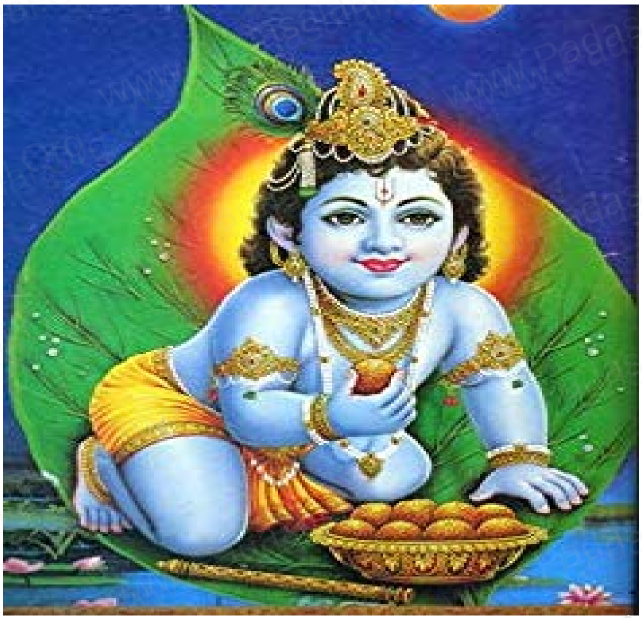
https://t.me/Padasalai_TET

- PGTRB - Group

https://t.me/Padasalai_PGTRB

- TNPSC - Group

https://t.me/Padasalai_TNPSC



SHRI KRISHNA ACADEMY

COMPUTER SCIENCE-MINIMUM MATERIAL 2019-2020

சிறப்பம்சங்கள்

- அரசுப் பொதுத்தேர்வு விடைக்குறிப்புகள் அடிப்படையில் கற்றல் கையேடு(Minimum Material) உருவாக்கப்பட்டுள்ளது.
- மெல்லக் கற்கும் மாணவர்கள் விரைவாகக் கற்று அதிக மதிப்பெண் பெறும் வகையில் இவ்விடைக்களஞ்சியம் அமைக்கப்பட்டுள்ளது.
- அரசுப் பொதுத்தேர்வு தேர்வில் கட்டாயமாகக் கேட்கப்படும் வினாக்களுக்கு(அகவய வினாக்கள்) விடைகள் கொடுக்கப்பட்டுள்ளது.
- கற்பதற்கு எளிய நடையில் வடிவமைக்கப்பட்டுள்ளது.
- புறவய வினாக்கள் மற்றும் அகவய வினாக்கள் விடைக்குறிப்புகளுடன் அமைந்துள்ளது.

CONTACT:9965531727,9443231727

XII COMPUTER SCIENCE

UNIT – 1 PROBLEM SOLVING TECHNIQUES

CHAPTER-1 FUNCTIONS

Part - I (1 Mark)

Choose the best answer:

1. The small sections of code that are used to perform a particular task is called
   (A) Subroutines (B) Files (C) Pseudo code (D) Modules
2. Which of the following is a unit of code that is often defined within a greater code structure?
   (A) Subroutines **(B) Function** (C) Files (D) Modules
3. Which of the following is a distinct syntactic block?
   (A) Subroutines (B) Function **(C) Definition** (D) Modules
4. The variables in a function definition are called as
   (A) Subroutines (B) Function (C) Definition **(D) Parameters**
5. The values which are passed to a function definition are called
   (A) Arguments (B) Subroutines (C) Function (D) Definition
6. Which of the following are mandatory to write the type annotations in the function definition?
   (A) Curly braces **(B) Parentheses** (C) Square brackets (D) indentations
7. Which of the following defines what an object can do?
   (A) Operating System (B) Compiler **(C) Interface** (D) Interpreter
8. Which of the following carries out the instructions defined in the interface?
   (A) Operating System (B) Compiler **(C) Implementation** (D) Interpreter
9. The functions which will give exact result when same arguments are passed are called
   (A) Impure functions (B) Partial Functions
   (C) Dynamic Functions **(D) Pure functions**
10. The functions which cause side effects to the arguments passed are called
   (A) impure function (B) Partial Functions
   (C) Dynamic Functions (D) Pure functions

Part - II (2 Marks)

Answer the following questions

1. **What is a subroutine?**
   - Subroutines are small sections of code .In Programming languages these subroutines are called as Functions.
2. **Define Function with respect to Programming language.**
   - A function is a unit of code that is often defined within a greater code structure.
3. **Write the inference you get from X: = (78).**
   - X:=(78) has an expression in it but (78) is not itself an expression. Rather, it is a function definition. Definitions bind values to names, in this case the value 78 being bound to the name 'X'.
4. **Differentiate interface and implementation.**

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

5. **Which of the following is a normal function definition and which is recursive function definition**

i) let rec sum x y:

return x + y

Ans:Recursive function definition

ii) let disp :

print 'welcome'

Ans: Normal function

iii) let rec sum num:

if (num!=0) then return num + sum (num-1)

else return num

Ans:Recursive function definition

Part - III (3 Marks)

Answer the following questions:

1. **Mention the characteristics of Interface.**
   - The class template specifies the interfaces to enable an object to be created and operated properly.
   - An object's attributes and behavior is controlled by sending functions to the object.
2. **Why strlen is called pure function?**
   - strlen is a pure function because the function takes one variable as a parameter, and accesses it to find its length.
   - This function reads external memory but does not change it, and the value returned derives from the external memory accessed.
3. **What is the side effect of impure function. Give example.**
   - A function has side effect when it has observable interaction with the outside world. Modifying the variable outside of function causes side effect.

Example:

```
let y:=0
(int)inc(int):x
y:=y+x;
return (y)
```

4. **Differentiate pure and impure function**

Pure Function	Impure Function
The return value solely depends on its arguments passed.	The return value does not solely depend on its arguments passed.
They do not modify the arguments which are passed to them	They may modify the arguments which are passed to them

5. **What happens if you modify a variable outside the function? Give an example.**
   - One of the most popular groups of side effects is modifying the variable outside of function.

For example

```
let y: = 0
(int) inc (int) x
y: = y + x;
return (y)
```

Part - III (5Marks)

Answer the following questions:

1. What are called Parameters and write a note on

(i) Parameter without Type (ii) Parameter with Type

Parameters (and arguments):

- Parameters are the variables in a function definition and arguments are the values which are passed to a function definition.

Parameter without Type:

Let us see an example of a function definition:

let rec pow a b:=

- The precondition **(requires)** and post condition **(returns)** of the function is given. Note we have not mentioned any types: **(data types)**.

Parameter with Type:

- Now let us write the same function definition with types for some reason

let rec pow (a: int) (b: int) : int :=

- When we write the type annotations for **'a'** and **'b'** the parentheses are mandatory.

2. Identify in the following program

```
let rec gcd a b :=
if b <> 0 then gcd b (a mod b) else return a
```

i) Name of the function

ii) Identify the statement which tells it is a recursive function

iii) Name of the argument variable

iv) Statement which invoke the function recursively

v) Statement which terminates the recursion

Ans:

- gcd
- rec
- a , b
- gcd b(a mod b)
- return a

3. Explain with example Pure and impure functions.

Pure functions

- Pure functions are functions which will give exact result when the same arguments are passed. For example the mathematical function sin (0) always results **0**.
- A function can be a pure function provided it should not have any external variable which will alter the behavior of that variable.
- Let us see an example

```
let square x
return: x * x
```

Impure functions:

- The variables used inside the function may cause side effects though the functions which are not passed with any arguments.
- In such cases the function is called impure function.

For example

```
let Random number
let a := random()
if a > 10 then
return: a
else
return: 10
```

Here the function Random is impure

4. Explain with an example interface and implementation.

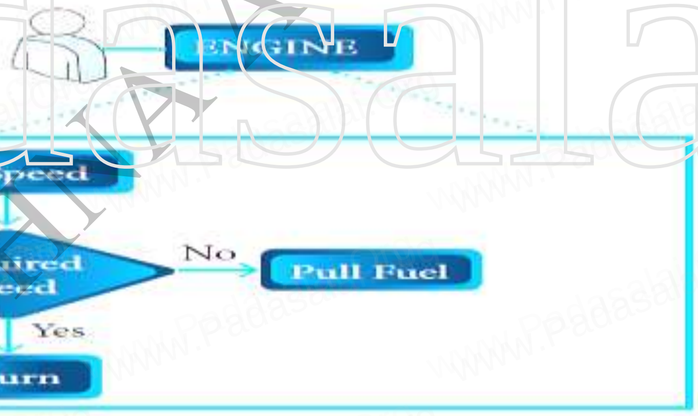
- An interface is a set of action that an object can do.
- In Object Oriented Programming language, an Interface is a description of all functions that a class must have in order to be a new interface.
- The interface defines an object's visibility to the outside world.

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

Characteristics of interface:

- The class template specifies the interfaces to enable an object to be created and operated properly.
- An object's attributes and behavior is controlled by sending functions to the object. For example, let's take the example of increasing a car's speed.



ADDITIONAL ONE MARK:

1. An object is also called ----------

   (a) instance (b) class (c) code (d) instruction

2. The ------ defines an objects visibility to the outside world

   (a) class **(b) interface** (c) implementation (d) program

3. Assertion : An interface is a set of action that an object can do.

   Reason : Interface defines an object's visibility to the outside world.

   a) Assertion is correct and reason is wrong **b) both are correct**

   c) Assertion is wrong and reason is correct d) both are wrong

4. How the object is processed and executed is the ---------

   (a) interface **(b) implementation** (c) class (d) program

5. Evaluation of ----------- functions does not cause any side effects to its output.

   (a) impure **(b) pure** (c) interface (d) class

6. -------- functions have side effects

   (a) impure (b) pure (c) interface (d) object

ADDITIONAL 2 MARK:

1. **Define interface.**
   - ❖ The interface defines an object's visibility to the outside world.
   - ❖ Interface just defines what an object can do, but won't actually do it

2. **Define implementation**

   Implementation carries out the instructions defined in the interface

ADDITIONAL 3 MARKS:

1. **Write note on pure function with example.**

   Pure functions are functions which will give exact result when the same arguments are passed.

   Example: **let square x**

   ```
   return: x * x
   ```

2. **Write about impure function.**

   The variables used inside the function may cause side effects though the functions which are not passed with any arguments.

   Example:

   ```
   let Random number
   let a := random()
       if a > 10 then
           return: a
       else
           return: 10
   ```

CHAPTER -2 DATA ABSTRACTION

Part - I (1 Mark)

Choose the best answer:

1. Which of the following functions that build the abstract data type ?
   (A) Constructors (B) Destructors (C) recursive (D)Nested
2. Which of the following functions that retrieve information from the data type?
   (A) Constructors **(B) Selectors** (C) recursive (D)Nested
3. The data structure which is a mutable ordered sequence of elements is called
   (A) Built in **(B) List** (C) Tuple (D) Derived data
4. A sequence of immutable objects is called
   (A) Built in (B) List **(C) Tuple** (D) Derived data
5. The data type whose representation is known are called
   (A) Built in datatype (B) Derived datatype
   (C) Concrete datatype (D) Abstract datatype
6. The data type whose representation is unknown are called
   (A) Built in datatype (B) Derived datatype
   (C) Concrete datatype **(D) Abstract datatype**
7. Which of the following is a compound structure?
   (A) Pair (B) Triplet (C) single (D) quadrate
8. Bundling two values together into one can be considered as
   (A) Pair (B) Triplet (C) single (D) quadrat
9. Which of the following allow to name the various parts of a multi-item object?
   (A) Tuples (B) Lists **(C) Classes** (D) quadrat
10. Which of the following is constructed by placing expressions within square brackets?
    (A) Tuples **(B) Lists** (C) Classes (D) quadrats

Part - II (2 Marks)

Answer the following questions:

1. **What is abstract data type?**
   - 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.

2. **Differentiate constructors and selectors.**

Constructor	Selector
To build the abstract data type.	To retrieve information from the data type.
Constructors create an object, bundling together different pieces of information	Selectors extract individual pieces of information from the object.

3. **What is a Pair? Give an example.**

- Any Way of bundling two values together into one can be considered as a pair.

  Example: lst:=[10,20]

4. **What is a List? Give an example.**

- List is constructed by placing expressions within square brackets separated by commas.

  Example for List is [10, 20].

5. **What is a Tuple? Give an example.**

- A tuple is a comma-separated sequence of values surrounded with parentheses. Tuple is similar to a list.

  Example: colour= ('red', 'blue', 'Green')

Part - III (3 Marks)

Answer the following questions:

1. **Differentiate Concrete data type and abstract data type.**

Concrete Data type	Abstract Data type
It direct implementations of a relatively simple concept.	(ADT's) offer a high level view of a concept independent of its implementation.
A concrete data type is a data type whose representation is known.	In abstract data type the representation of a data type is unknown.

2. **Which strategy is used for program designing? Define that Strategy.**
   - **Wishful Thinking** strategy is used for program designing.
3. **Identify Which of the following are constructors and selectors?**

   (a) N1=number() (b) accetnum(n1) (c) displaynum(n1)

   (d) eval(a/b) (e) x,y= makeslope (m), makeslope(n) (f) display()

   Ans:

   (a), (e) → constructors

   (b), (d), (c),(d) → selectors

4. **What are the different ways to access the elements of a list. Give example.**
   - Two ways:

     (i)multiple assignment,
   - **Example:**

     lst := [10, 20]

     (ii)A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.
   - **Example:**

     lst[0]

     10

5. **Identify Which of the following are List, Tuple and class ?**

   (a) arr [1, 2, 34] (b) arr (1, 2, 34) (c) student [rno, name, mark]

   (d) day= ('sun', 'mon', 'tue', 'wed') (e) x= [2, 5, 6.5, [5, 6], 8.2]

   (f) employee [eno, ename, esal, eaddress]

   Ans:

   (a), (e) → list

   (b), (d) → tuple

   (c), (f) → class

Part - IV (5Marks)

Answer the following questions:

1. **How will you facilitate data abstraction. Explain it with suitable example**
   - To create two types of functions: constructors and selectors.

   Constructors and Selectors:
   - Constructors are functions that build the abstract data type.

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

Example:

city = makecity (name, lat, lon)

To extract the information of a city object, you would use functions like

getname(city)

getlat(city)

getlon(city)

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

2. What is a List? Why List can be called as Pairs. Explain with suitable example

- List is constructed by placing expressions within square brackets separated by commas. Such an expression is called a list literal. List can store multiple values. Each value can be of any type and can even be another list.

Example for List is [10, 20].

lst := [10, 20] , y := lst

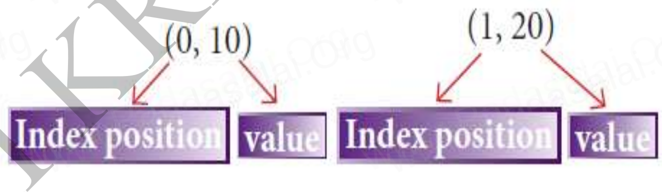
- A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.

lst[0]

10

In both the example mentioned above mathematically we can represent list similar to a set.

lst[(0, 10), (1, 20)] – where



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.

Representing Rational Numbers Using List

You can now represent a rational number as a pair of two integers in

pseudo code : a numerator and a denominator.

rational(n, d):

return [n, d]

```
numer(x):
return x[0]
denom(x):
return x[1]
```

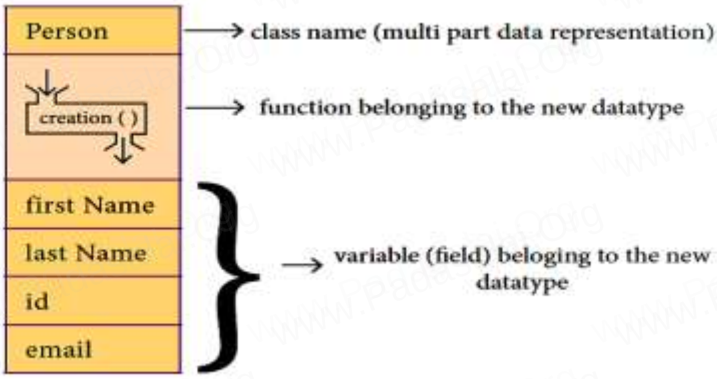
3. How will you access the multi-item. Explain with example.

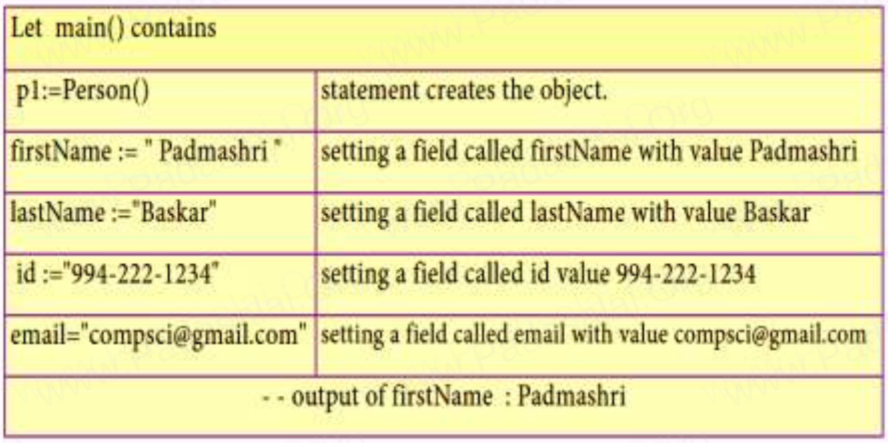
- We can use the structure construct (In OOP languages it's called class construct) to represent multi-part objects where each part is named (given a name). Consider the following pseudo code:

```
class Person:
creation( )
firstName := " "
lastName := " "
id := " "
email := " "
```

- The new data type Person is pictorially represented as
- Consider the following pseudo code:

```
class Person:
creation( )
firstName := "  "
lastName := "  "
id := "  "
email := "  "
```


Person
class name (multi part data representation)
creation ( )
function belonging to the new datatype
first Name
last Name
id
email
variable (field) beloging to the new datatype



Let main() contains	
p1:=Person()	statement creates the object.
firstName := " Padmashri "	setting a field called firstName with value Padmashri
lastName :="Baskar"	setting a field called lastName with value Baskar
id :="994-222-1234"	setting a field called id value 994-222-1234
email="compsci@gmail.com"	setting a field called email with value compsci@gmail.com
- - output of firstName : Padmashri	

ADDITIONAL ONE MARK:

1. Python provides a compound structure called ------

   (a) list **(b) pair** (c) Tuple (d) none

2. An expression in the list is called ------

   (a) pair (b) element **(c) list literal** (d) data

3. ----- can store multiple values

   (a) Tuple **(b) list** (c) element (d) pair

4. There are ------ ways to access the elements from the list

   (a) 3 **(b) 2** (c) 4 (d) 6

5. Any way of bundling two values together into one can be considered as a -------

   (a) list **(b) pair** (c) Tuple (d) data

6. ------ is a compound data type that holds two other pieces of data

   (a) list **(b) pair** (c) class (d) Tuple

7. In which, cannot change the elements, once it is assigned

   (a) Tuple (b) pair (c) list (d) element

8. In ------ we can change the lements.

   (a) Tuple **(b) list** (c) pair (d) element

9. Choose the correct statements.

   a)Constructors are functions that build the abstract data type.

   b) a class that not includes the data and functions.

   c) list is called array d) We can change the duple elements

10.------- as bundled data and the functions that work on that data.

   (a) object **(b) class** (c) interface (d) list

ADDITIONAL 3 MARKS:

1. **Define class**

   A class as bundled data and the functions that work on that data.

2. **What are the two parts of a program? explain**

   The two parts of a program are, the part that operates on abstract data and the part that defines a concrete representation,

CHAPTER-3 SCOPING

Part - I (1 Mark)

Choose the best answer:

1. Which of the following refers to the visibility of variables in one part of a program to another part of the same program?

   (A) Scope (B) Memory (C) Address (D) Accessibility

2. The process of binding a variable name with an object is called

   (A) Scope **(B) Mapping** C) late binding (D) early binding

3. Which of the following is used in programming languages to map the variable and object?

   (A) :: (B) := **(C) =** (D) ==

4. Containers for mapping names of variables to objects is called

   (A) Scope (B) Mapping (C) Binding **(D) Namespaces**

5. Which scope refers to variables defined in current function?

   (A) Local Scope (B) Global scope (C) Module scope (D) Function Scope

6. The process of subdividing a computer program into separate sub-programs is called

   (A) Procedural Programming **(B) Modular programming**

   (C)Event Driven Programming (D) Object oriented Programming

7. Which of the following security technique that regulates who can use resources in a computing environment?

   (A) Password (B)Authentication **(C) Access control** (D) Certification

8. Which of the following members of a class can be handled only from within the class?

   (A) Public members (B)Protected members

   (C) Secured members **(D) Private members**

9. Which members are accessible from outside the class?

(A) Public members (B)Protected members

(C) Secured members (D) Private members

10. The members that are accessible from within the class and are also available to its sub-classes is called

(A) Public members **(B)Protected members**

(C) Secured members (D) Private members

Part - II (2 Marks)

Answer the following questions:

1. **What is a scope?**
    - Scope refers to the visibility of variables,
2. **Why scope should be used for variable. State the reason.**
    - Every variable defined in a program has global scope. Once defined, every part of the program can access that variable. changes inside the function can't affect the variable on the outside of the function in unexpected ways.
3. **What is Mapping?**
    - The process of binding a variable name with an object is called mapping.
4. **What do you mean by Namespaces?**
    - Namespaces are containers for mapping names of variables to objects.

      Example: name:=object
5. **How Python represents the private and protected Access specifies?**
    - Python prescribes a convention of prefixing the name of the variable with single or double underscore to emulate the behavior of protected and private access specifies.

Part - III (3 Marks)

Answer the following questions:

1. **Define Local scope with an example.**
    - Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope.

Example

```
  Disp():
 :=7
 print a
  Disp()
```

2. **Define Global scope with an example.**

- A variable which is declared outside of all the functions in a program is known as global variable.

Example

```
a:=10
Disp():
a:=7
print a
```

2. **Define Enclosed scope with an example.**

- A function (method) with in another function is called nested function.

Example

```
1. Disp():
2. a:=10
3. Disp1():
4. print a
5. Disp1()
6. print a
```

Output of the Program

```
10
 10
```

4. **Why access control is required?**

- Access control is a security technique that regulates who or what can view or use resources in a computing environment.

3. **Identify the scope of the variables in the following pseudo code and write its output**

```
color:= Red
mycolor():
b:=Blue
myfavcolor():
g:=Green
printcolor, b, g
myfavcolor()
printcolor, b
mycolor()
print color
```

Ans: b:=Blue →local scope

g:=Green →enclosed scope

color:=Red →global scope

Output: Blue

```
Green
Blue
Red
```

Part – IV (5Marks)

Answer the following questions:

1. **Explain the types of scopes for variable or LEGB rule with example.**

Types of Variable Scope:

Local Scope

Local scope refers to variables defined in current function. Only if it does not find it there, the outer scopes are checked.

Example

```
Disp():
a:=7
print a
Disp()
```

Global Scope

A variable which is declared outside of all the functions in a program is known as global variable.

Example

```
1. a:=10
2. Disp():
3. a:=7
4. print a
```

Enclosed Scope

- A variable which is declared inside a function which contains another function definition with in it,

Example

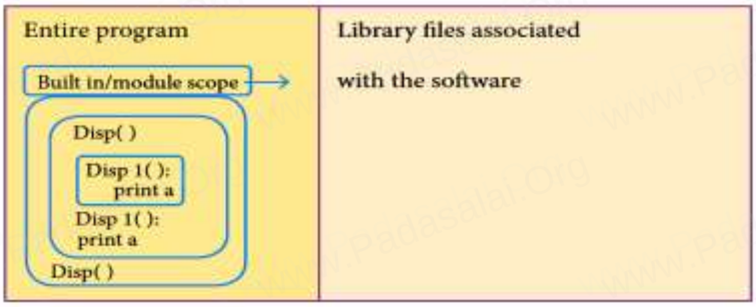
```
1. Disp():
2. a:=10
3. Disp1():
4. print a
5. Disp1()
6. print a
7. Disp()
```

Output of the Program

```
10
10
```

Built-in Scope

- Finally, we discuss about the widest scope. The built-in scope has all the names that are pre-loaded into the program scope.



(OR)

LEGB rule:

- Scope also defines the order in which variables have to be mapped to the object in order to obtain the value.:
- The **LEGB** rule is used to decide the order in which the scopes are to be searched for scope resolution.

Local(**L**)	-	Defined inside function/class
Enclosed(**E**)	-	Defined inside enclosing functions
Global(**G**)	-	Defined at the uppermost level
Built-in (**B**)	-	Reserved names in built-in functions (modules)

2. Write any Five Characteristics of Modules.

The following are the desirable characteristics of a module.

- Modules contain instructions, processing logic, and data.
- Modules can be separately compiled and stored in a library.
- Modules can be included in a program.
- Module segments can be used by invoking a name and some parameters.
- Module segments can be used by other modules.

3. Write any five benefits in using modular programming.

- Less code to be written.
- The code is stored across multiple files.
- Code is short, simple and easy to understand.
- The same code can be used in many applications.
- The scoping of variables can easily be controlled.

ADDITIONAL ONE MARK:

1. Every variable defined in a program has ------ scope.

   (a) local **(b) global** (c) variable (d) class

2. ----- are containers for mapping names of variables to object.

   (a) value (b) variable (c) memory **(d) namespaces**

3. ------ defined inside function class

   (a) global **(b) local** (c) built in (d) enclosed

4. A function with another function is called ------ function.

   (a) nested (b) upper (c) local (d) global

5. Choose the correct statement:

   a) Namespaces are contains for mapping names of variables to object.

   b) Scope refers to functions.

   c) LEGB role is used to decide the orders in scope of variables.

   d) There are 5 types of scope.

   a) a, c is correct b) a, b, c is correct c) a, d is correct d) only a is correct

CHAPTER- 4 ALGORITHMIC STRATEGIES

Part - I (1 Mark)

Choose the best answer:

1. The word comes from the name of a Persian mathematician Abu Jafar Mohammed ibn Musa al Khowarizmi is called?

   (A) Flowchart (B) Flow **(C) Algorithm** (D) Syntax

2. From the following sorting algorithms which algorithm needs the minimum number of swaps?

   (A) Bubble sort (B) Quick sort (C) Merge sort **(D) Selection sort**

3. Two main measures for the efficiency of an algorithm are

   (A) Processor and memory (B) Complexity and capacity

   (C) Time and space (D) Data and space

4. The complexity of linear search algorithm is

   (A) O(n) (B) O(log n) (C) O(n2) (D) O(n log n)

5. From the following sorting algorithms which has the lowest worst case complexity?

   (A) Bubble sort B) Quick sort (C) Merge sort (D) Selection sort

6. Which of the following is not a stable sorting algorithm?

   (A) Insertion sort **(B) Selection sort** (C) Bubble sort (D) Merge sort

7. Time complexity of bubble sort in best case is

   (A) θ (n) (B) θ (nlogn) (C) θ (n2) (D) θ (n(logn) 2)

8. The Θ notation in asymptotic evaluation represents

   (A) Base case **(B) Average case** (C) Worst case (D) NULL case

9. If a problem can be broken into sub problems which are reused several times, the problem possesses which property?

   (A) Overlapping sub problems (B) Optimal substructure

   (C) Memoization (D) Greedy

10.In dynamic programming, the technique of storing the previously calculated values is called?

(A) Saving value property (B) Storing value property

(C) Memoization (D) Mapping

Part - II (2 Marks)

Answer the following questions:

1. **What is an Algorithm?**
   - It is a step-by-step procedure for solving a given problem.
2. **Define Pseudo code.**
   - Pseudo code is an informal way of programming language syntax.
3. **Who is an Algorist?**
   - A person skilled in the technique of performing basic decimal arithmetic, know as algorism.
3. **What is Sorting?**
   - Arranging data in ascending or descending order is called sorting.
4. **What is searching? Write its types.**
   - Searching is the process of finding a particular data in a collection of data.

Types:

(i)Linear (or) sequential search (ii)Binary (or) half interval search

Part - III (3 Marks)

Answer the following **questions:**

1. **List the characteristics of an algorithm.**

   (i)Input (ii)Output (iii)Finiteness (iv)Definiteness (v)Effectiveness (vi)Correctness Portable

2. **Discuss about Algorithmic complexity and its types.**

   Time Complexity: The Time complexity of an algorithm is given by the number of steps taken by the algorithm to complete the process.

   Space Complexity: Space complexity of an algorithm is the amount of memory required to run to its completion.

   - fixed part
   - Variable part

3. What are the factors that influence time and space complexity. (*)

Time Factor -Time is measured by counting the number of key operations like comparisons in the sorting algorithm.

Space Factor - Space is measured by the maximum memory space required by the algorithm.

4. Write a note on asymptotic notation.

Asymptotic Notations are languages that uses meaningful statements about time and space complexity.

(i) Big O:

(ii) Big Ω:

(iii) Big ϴ :

5.What do you understand by Dynamic programming?

- Dynamic programming is an algorithmic design method that 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.

Part - IV (5Marks)

Answer the following questions:

1. Explain the characteristics of an algorithm. (Any 5)

Input	Zero or more quantities to be supplied
Output	At least one quantity is produce.
Finiteness	Algorithms must terminate after finite number of steps.
Effectiveness	Every instruction must be carried out effectively
Correctness	The algorithms should be error free.
Simplicity	Easy to implement
Feasibility	Should be feasible with the available resources.

2. Discuss about Linear search algorithm.

Linear Search

Linear search also called sequential search is a sequential method for finding a particular value in a list.

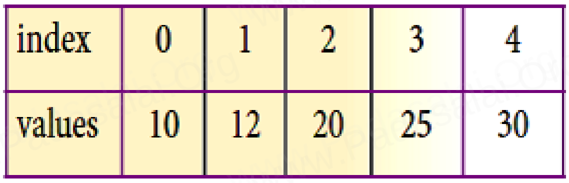
Pseudo code

(i) Traverse the array using for loop

(ii)In every iteration, compare the target search key value with the current value of the list.

(iii) If no match is found, display the search element not found.

- To search the number 25 in the array given below, In this example number 25 is found at index number 3.



index	0	1	2	3	4
values	10	12	20	25	30

Example 1:

Input: values[] = {5, 34, 65, 12, 77, 35}

target = 77

Output: 4

3. What is Binary search? Discuss with example.

- Binary search also called half-interval search algorithm.
- The binary search algorithm can be done as divide-and-conquer search algorithm and executes in logarithmic time.

Binary Search Working principles

- First, we find index of middle element of the array by using this formula :

mid = low + (high - low) / 2

- 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

- The search element still not found hence we calculate the mid again by using the formula

high=mid-1

mid=low+(High-low)/2

4. Explain the Bubble sort algorithm with example.

Bubble sort algorithm :

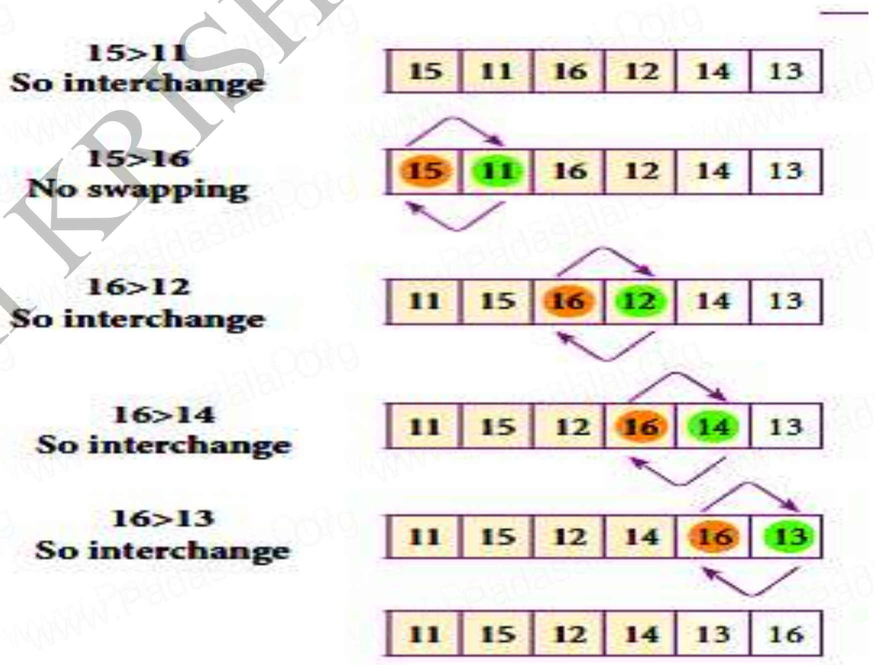
- Bubble sort is a simple sorting algorithm. The algorithm starts at the beginning of the list of values stored in an array.
- It compares each pair of adjacent elements and swaps them if they are in the unsorted order. This comparison and passed to be continued until no swaps are needed, which indicates that the list of values stored in an array is sorted.

Pseudo code

- Start with the first element i.e., index = 0, compare the current element with the next element of the array.
- If the current element is greater than the next element of the array, swap them.
- If the current element is less than the next or right side of the element, move to the next element. Go to Step 1 and repeat until end of the index is reached.

Example : Let's consider an array with values {15, 11, 16, 12, 14, 13}

Below, we have a pictorial representation of how bubble sort will sort the given array.



5. Explain the concept of Dynamic programming with suitable example.

- Dynamic programming is an algorithmic design method that 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.

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. Fib 0 & Fib 1.
- Fibonacci series satisfies the following conditions:

  Fib n = Fibn-1 + Fibn-2

- Hence, a Fibonacci series for the n value 8 can look like this

  Fib 8 = 0 1 1 2 3 5 8 13

ADDITIONAL ONE MARK:

1. ------ is a step-by-step procedure for solving a given problem.

   (a) program **(b) algorithm** (c) instruction (d) data

2. Pick out odd one from the following

   (a) array (b) structure (c) Tuples **(d) information**

3. ------- is to delete an existing item is in a data structure

   (a) sort **(b) delete** (c) insert (d) search

4. ----- are generic and not limited to computer alone.

   (a) algorithm (b) program (c) data (d) instruction

5. ------ is an optimization technique used primarily to speed up computer programs.

   (a) memorization (b) divide and conquer (c) pseudo code (d) Binary search

ADDITIONAL 3 MARKS:

1. **Define algorithm analysis.**

   An estimation of the time and space complexities of an algorithm for varying input sizes is called algorithm analysis.

2. **What are the two components of space complexity? Explain.**

   A fixed part: It is defined as the total space required to store certain data and variables for an algorithm. For example, simple variables and constants used in an algorithm.

   A variable part: It is defined as the total space required by variables, which sizes depends on the problem and its iteration. For example: recursion used to calculate factorial of a given value n.

3. **What is meant by best algorithm explain.**

   The best algorithm to solve a given problem is one that requires less space in memory and takes less time to execute its instructions to generate output.