



COMPUTER SCIENCE

12th Standard

Based on the Updated New Textbook

Salient Features

- ☞ Answers for all Textual Questions.
- ☞ Exhaustive Additional MCQs, VSA and SA question with answers are given in each chapter.
- ☞ All the objective type (1 Mark) questions are given with 4 options.
 - (i) Choosing the correct option
 - (ii) Matching
 - (iii) Filling the blanks
 - (iv) Choosing the Correct \ Incorrect Statement.
 - (v) Picking the Odd one Out.
 - (vi) Assertion and Reason.
- ☞ Model Question Papers 1 to 6 (PTA) : Questions are incorporated in the appropriate sections.
- ☞ Govt. Model Question Paper - 2019 (Govt. MQP-2019), Quarterly Exam - 2019, 2023 & 2024 (QY-2019, '23 & '24), Half Yearly Exam - 2019 & 2023 (HY-2019 & '23), Public Exam March-2020, May-2022, March-2023, 2024 & 2025 (Mar-2020, May- 2022, Mar-'23, '24 & '25), Govt. Supplementary Exam September-2020, August-2021, July-2022, June-2023 & 2024 (Sep-2020, Aug-'21, July-'22, June-'23, '24 & '25) and First & Second Revision Test - 2022 (FRT & SRT-'22) questions are incorporated in the appropriate sections.
- ☞ Instant Supplementary Exam June - 2025 & Quarterly Exam - 2025 Question Paper are given with answers.



SURA PUBLICATIONS

Chennai

For Orders Contact



**80562 94222 / 81242 01000 / 81243 01000
96001 75757 / 98409 26027**

All rights reserved © SURA Publications.

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, digitally, electronically, mechanically, photocopying, recorded or otherwise, without the written permission of the publishers. Strict action will be taken.

Code No : SG 91

Author :

Mr. Shanmugasundaram

(Post Graduate Teacher, Chennai)

Edited by :

Mrs. Malathy Krishnamoorthy M.Sc.,

Erode

Reviewed by :

Mr. Balaji M.Sc., M.Phil.

Chennai

Also available for XI & XII Standard

Guides :

- ✦ சுராவின் தமிழ் உரைநூல்
- ✦ Sura's Smart English
- ✦ Sura's Mathematics (EM/TM)
- ✦ Sura's Physics (EM/TM)
- ✦ Sura's Chemistry (EM/TM)
- ✦ Sura's Bio-Botany & Botany (EM/TM)
(Short Version & Long Version)

- ✦ Sura's Bio-Zoology & Zoology (EM/TM)
(Short Version & Long Version)
- ✦ Sura's Computer Science (EM/TM)
- ✦ Sura's Computer Applications (EM/TM)
- ✦ Sura's Commerce (EM/TM)
- ✦ Sura's Economics (EM/TM)
- ✦ Sura's Accountancy (EM/TM)
- ✦ Sura's Business Maths (EM)

Head Office:

Sura Publications

1620, 'J' Block, 16th Main Road,
Anna Nagar, Chennai - 600 040.

Phone : 044 - 4862 9977, 044 - 4862 7755.

E-mail : orders@surabooks.com

Website : www.surabooks.com

For Orders Contact



80562 94222
81242 01000
81243 01000
96001 75757
98409 26027

PREFACE

“ The woods are lovely, dark and deep. “

But I have promises to keep, and

miles to go before I sleep

- Robert Frost

Respected Principals, Correspondents, Head Masters / Head Mistresses, Teachers,

From the bottom of our heart, we at SURA Publications sincerely thank you for the support and patronage that you have extended to us for more than a decade.

It is in our sincerest effort we take the pride of releasing **SURA’s Computer Science Guide** for +2 Standard. This guide has been authored and edited by qualified teachers having teaching experience for over a decade in their respective subject fields. This Guide has been reviewed by reputed Professors who are currently serving as Head of the Department in esteemed Universities and Colleges.

With due respect to Teachers, I would like to mention that this guide will serve as a teaching companion to qualified teachers. Also, this guide will be an excellent learning companion to students with exhaustive exercises and in-text questions in addition to precise answers for textual questions.

In complete cognizance of the dedicated role of Teachers, I completely believe that our students will learn the subject effectively with this guide and prove their excellence in Board Examinations.

I once again sincerely thank the Teachers, Parents and Students for supporting and valuing our efforts.

God Bless all.

Subash Raj, B.E., M.S.

- Publisher

Sura Publications

All the Best

TO ORDER WITH US

SCHOOLS and TEACHERS:

We are grateful for your support and patronage to '**SURA PUBLICATIONS**'

Kindly prepare your order in your School letterhead and send it to us.

For Orders contact: 81242 01000 / 81243 01000

DIRECT DEPOSIT

A/c Name : **Sura Publications**
Our A/c No. : **36550290536**
Bank Name : **STATE BANK OF INDIA**
Bank Branch : Padi
IFSC : SBIN0005083

A/c Name : **Sura Publications**
Our A/c No. : **21000210001240**
Bank Name : **UCO BANK**
Bank Branch : Anna Nagar West
IFSC : UCBA0002100

A/c Name : **Sura Publications**
Our A/c No. : **6502699356**
Bank Name : **INDIAN BANK**
Bank Branch : Asiad Colony
IFSC : IDIB000A098

A/c Name : **Sura Publications**
Our A/c No. : **1154135000017684**
Bank Name : **KVB BANK**
Bank Branch : Anna Nagar
IFSC : KVBL0001154

A/c Name : **Sura Publications**
Our A/c No. : **13240200032412**
Bank Name : **FEDERAL BANK**
Bank Branch : Anna Nagar
IFSC : FDRL0001324

A/c Name : **Sura Publications**
Our A/c No. : **50200031530945**
Bank Name : **HDFC BANK**
Bank Branch : Cenotaph Road, Teynampet
IFSC : HDFC0001216

After Deposit, please send challan and order to our address.

email to : orders@surabooks.com / Whatsapp : 81242 01000.



For Google Pay :
98409 26027



For PhonePe :
98409 26027



DEMAND DRAFT / CHEQUE

Please send Demand Draft / cheque in favour of '**SURA PUBLICATIONS**' payable at **Chennai**. The Demand Draft / cheque should be sent with your order in School letterhead.



SURA PUBLICATIONS

1620, 'J' Block, 16th Main Road, Anna Nagar, Chennai - 600 040.

Phone : 044-4862 9977, 044-4862 7755.

Mobile : 96001 75757 / 81242 01000 / 81243 01000.

email : orders@surabooks.com Website : www.surabooks.com

CONTENTS

Unit	Chapter No	Title	Page No	Month
UNIT- I Problem Solving Techniques	1.	Function	1-7	June
	2.	Data Abstraction	8-15	
	3.	Scoping	16-24	
	4.	Algorithmic Strategies	25-38	
UNIT- II Core Python	5.	Python -Variables and Operators	39-53	July
	6.	Control Structures	54-69	
	7.	Python functions	70-89	
	I MID TERM TEST (June, July)			
	8.	Strings and String manipulation	90-104	August
UNIT-III Modularity and OOPS	9.	Lists, Tuples, Sets and Dictionary	105-128	
	10.	Python Classes and objects	129-140	
MONTHLY TEST (August Portion)				
QUARTERLY EXAMINATION (June to September)				

Unit	Chapter No	Title	Page No	Month
UNIT-IV Database concepts and MySQL	11.	Database Concepts	141-156	October
	12.	Structured Query Language (SQL)	157-176	
	13.	Python and CSV files	177-194	
UNIT-V Integrating Python with MySQL and C++	14.	Importing C++ programs in Python	195-208	November
	MONTHLY TEST (October Portion)			
	15.	Data manipulation through SQL	209-222	
	16.	Data visualization using pyplot: line chart, pie chart and bar chart	223-234	
II MID TERM TEST (October & November)				
MONTHLY TEST (November Portion)				
HALF YEARLY EXAMINATION (Full Portion)				

Instant Supplementary Exam June - 2025 & Quarterly Exam - 2025
Question Paper with answers

235-250

UNIT-I PROBLEM SOLVING TECHNIQUES

Chapter-1 FUNCTION

CHAPTER SNAPSHOT

1.1 Introduction	1.3.1 Characteristics of interface
1.2 Function with respect to Programming language	1.4 Pure functions
1.2.1 Function Specification	1.4.1 Impure functions
1.2.2 Parameters and arguments	1.4.2 Side-effects (Impure functions)
1.3 Interface Vs Implementation	1.4.3 Chameleons of Chromeland problem using function

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

- The small sections of code that are used to perform a particular task is called
[Aug-2021; FRT-'22; Mar & HY-'23; QY-'24]
(a) Subroutines (b) Files
(c) Pseudo code (d) Modules
[Ans. (a) Subroutines]
- Which of the following is a unit of code that is often defined within a greater code structure?
[July-'22]
(a) Subroutines (b) Function
(c) Files (d) Modules
[Ans. (b) Function]
- Which of the following is a distinct syntactic block?
[PTA-6; FRT & May-'22; Mar-'25]
(a) Subroutines (b) Function
(c) Definition (d) Modules
[Ans. (c) Definition]
- The variables in a function definition are called as
[PTA-2; QY-2019 & '23; June-'23]
(a) Subroutines (b) Function
(c) Definition (d) Parameters
[Ans. (d) Parameters]
- The values which are passed to a function definition are called [HY-2019; FRT-'22; June-'24]
(a) Arguments (b) Subroutines
(c) Function (d) Definition
[Ans. (a) Arguments]
- Which of the following are mandatory to write the type annotations in the function definition?
[PTA-4; FRT-'22]
(a) { } (b) ()
(c) [] (d) < > [Ans. (b) ()]
- Which of the following defines what an object can do?
[Mar-'24]
(a) Operating System (b) Compiler
(c) Interface (d) Interpreter
[Ans. (c) Interface]
- Which of the following carries out the instructions defined in the interface?
(a) Operating System (b) Compiler
(c) Implementation (d) Interpreter
[Ans. (c) Implementation]
- The functions which will give exact result when same arguments are passed are called
[PTA-3; Mar.-2020; June-'25]
(a) Impure functions (b) Partial Functions
(c) Dynamic Functions (d) Pure functions
[Ans. (d) Pure functions]
- The functions which cause side effects to the arguments passed are called
(a) Impure function (b) Partial Functions
(c) Dynamic Functions (d) Pure functions
[Ans. (a) Impure function]

S. No.	Pure	Impure
(iii)	They do not have any side effects.	They have side effects. For example: random(), Date().
(iv)	They do not modify the arguments which are passed to them	They may modify the arguments which are passed to them

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. What are called Parameters and write a note on [PTA-2; May-'22; June-'24 & '25; QY-'24]

(i) Parameter without Type [FRT-'22]

(ii) Parameter with Type

Ans. Parameters and arguments : Parameters are the variables used in a function definition. Arguments are the values passed to those parameters when the function is called.

(i) **Parameter without Type :** Let us see an example of a function, definition :

(requires: $b \geq 0$)

(returns: a to the power of b)

let rec pow a b:=

if b=0 then 1

else a * pow a (b-1)

- 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 (**requires**) and postcondition (**returns**) of the function is given..
- In this function, we have not mentioned any data types..
- Some compilers can infer types automatically, while others require the programmer to specify them.
- In the above example:
 - The if expression returns 1 (an integer), so the whole expression has type int.
 - Therefore, the function's return type is int.
 - Since b is compared with 0, it must be of type int.
 - Because a is multiplied using *, it must also be of type int.

(ii) **Parameter with Type :** Now, let's define the same function with types:

(requires: $b \geq 0$)

(returns: a to the power of b)

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

if b=0 then 1

else a * pow a (b-1)

- When writing type annotations for 'a' and 'b', parentheses are mandatory..
- Usually, we can omit type annotations because the compiler can infer them.
- However, adding types explicitly can help debug type errors more easily.
- The keyword let introduces a function definition.
- To define a recursive function, use let rec instead of just let.

Syntax:

let rec fn a1 a2 ... an := k

- fn – function name
- a1 to an – parameters
- rec – required if the function calls itself (recursive function)
- Here the 'fn' is used as a function name. The names 'a1' to 'an' are variables used as parameters. The keyword 'rec' is required if 'fn' is to be a recursive function; otherwise it may be omitted.

2. Identify in the following program

```
let rec gcd a b :=
```

```
if b <> 0 then gcd b (a mod b) else return a
```

- Name of the function
- Identify the statement which tells it is a recursive function
- Name of the argument variable
- Statement which invoke the function recursively
- Statement which terminates the recursion

[PTA-5; HY-'23]

Ans. (i) gcd

(ii) let rec gcd

(iii) a, b

(iv) gcd b (a mod b)

(v) return a

3. Explain with example Pure and impure functions. [Mar-'25]

Ans. Pure functions :

- Pure functions give exact results when the same arguments are passed..

- (ii) Example: The mathematical function $\sin(0)$ always results in 0.
- (iii) A pure function should not have any external variable that alters its behavior.

(iv) **Example :**

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

Here, square is pure because it gives the same result for the same input.

- (v) One advantage is that if a function is pure, then if it is called several times with the same arguments, the compiler only needs to actually call the function once..

Impure functions :

- (i) 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..
- (ii) When a function depends on variables or functions outside of its definition block, you can never be sure that the function will behave the same every time it's called.

(iii) **Example :**

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

- (iv) Here, random() is impure because it gives different results for the same function call.

4. Explain with an example interface and implementation. [QY-'23]

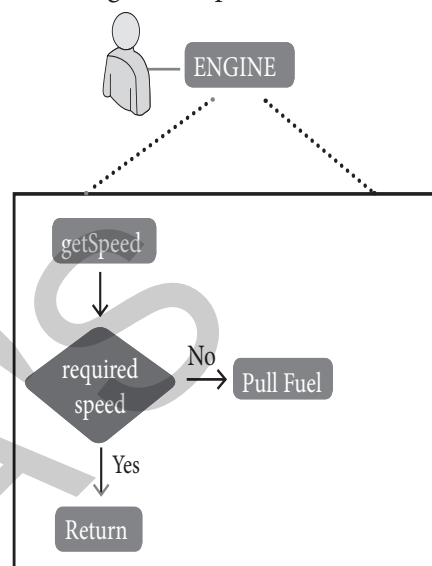
Ans. Interface :

- (i) An interface is a set of action that an object can do. For example when you press a light switch, the light goes on, you may not have cared how it splashed the light. In Object Oriented Programming language, an Interface is a description of all functions.
- (ii) In our example, anything that "ACTS LIKE" a light, should have function definitions like turn_on () and a turn_off (). The purpose of interfaces is to allow the computer to enforce the properties of the class.

Implementation :

- (i) Implementation carries out the instructions defined in the interface.
- (ii) How the object is processed and executed is the implementation.

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



- (iv) The person driving a car doesn't need to know how the engine works. To increase speed, the driver simply presses the accelerator to get the desired result. Here, the accelerator acts as the interface between the driver (the calling/invoking object) and the engine (the called object).
- (v) For example, the function call could be Speed(70):This is the interface.
- (vi) Inside the engine, fuel, air, pressure, and electricity work together to create power. These internal actions are hidden from the driver, who only wants to go faster.
- (vii) Thus, the interface (accelerator) is separated from the implementation (engine working).

HANDS ON PRACTICE

1. Write algorithmic function definition to find the minimum among 3 numbers.

Ans. let min x y z :=
 if x < y then
 if x < z then x else z
 else
 if y < z then y else z

2. Write algorithmic recursive function definition to find the sum of n natural numbers.

Ans. let rec sum num:
 if (num!=0) then return num+sum(num-1)
 else
 return num

PTA QUESTIONS AND ANSWERS

1 MARK

1. A function definition which call itself : [PTA-1]
 (a) Pure function (b) Impure function
 (c) Normal function
 (d) Recursive function

[Ans. (d) Recursive function]

3 MARKS

1. Write a function that finds the minimum of its three arguments. [PTA-4; QY-2019]

Ans. let min x y z :=
 if x < y then
 if x < z then x else z
 else
 if y < z then y else z

GOVERNMENT EXAM QUESTIONS AND ANSWERS

1 MARK

1. Which is the basic building block of computer programs? [Sep-2020]
 (a) Argument (b) Parameter
 (c) Subroutine (d) Interface
 [Ans. (c) Subroutine]

2 MARKS

1. Define pure function. Give one example. [QY-2019]

Ans. 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.

ADDITIONAL QUESTIONS AND ANSWERS

CHOOSE THE CORRECT ANSWER 1 MARK

1. Which of the following are expressed using statements of a programming language?
 (a) Functions (b) Algorithm
 (c) Interface (d) Implementation
 [Ans. (b) Algorithm]

2. What must be used when a bulk of statements to be repeated for many number of times?

(a) Algorithm (b) Program
 (c) Subroutines (d) Parameters

[Ans. (c) Subroutines]

3. Which of the following contains a set a code that works an many kinds of input and produces a concrete output?

(a) Function (b) Algorithm
 (c) Arguments (d) Language

[Ans. (a) Function]

4. Which of the following are the values which are passed to a function definition?

(a) Parameters (b) Algorithm
 (c) Data types (d) Arguments

[Ans. (d) Arguments]

5. The function definition is introduced by the keyword

(a) def (b) rec
 (c) let (d) infer

[Ans. (c) let]

6. The recursive function is defined using the keyword

(a) let (b) let rec
 (c) name (d) infer

[Ans. (b) let rec]

7. Which of the following is a description of all functions in object oriented programming language?

(a) Implementation (b) parameter
 (c) Interface (d) Arugument

[Ans. (c) Interface]

8. Which of the following is an instance created from the class?

(a) parameter (b) function
 (c) subroutines (d) object

[Ans. (d) object]

9. Which of the following is an example of impure function?

(a) Strlen() (b) random()
 (c) sqrt() (d) pure()

[Ans. (b) random()]

10. In which type of function the return type is solely depends on its argument passed?

(a) pure (b) impure
 (c) parameterized (d) monochromatize

[Ans. (a) pure]

11. In which type of function the return type does not solely depends on its argument passed?

- (a) Pure (b) Parameterized
(c) Impure (d) Monochromatize

[Ans. (c) Impure]

MATCH THE FOLLOWING

1. Match the following function definitions with their terms.

let rec odd xy : =

	List I		List II
i)	Keyword	1)	Xy
ii)	Recursion	2)	Odd
iii)	Function name	3)	Rec
iv)	Parameters	4)	let

- | | | | | |
|-----|-----|------|-------|------|
| | (i) | (ii) | (iii) | (iv) |
| (a) | 4 | 3 | 2 | 1 |
| (b) | 1 | 2 | 3 | 4 |
| (c) | 4 | 1 | 2 | 3 |
| (d) | 1 | 4 | 2 | 3 |

[Ans. (a) (i)-4; (ii)-3; (iii)-2; (iv)-1]

CHOOSE AND FILL IN THE BLANKS

1. Subroutines are called as _____

- (a) Algorithm (b) Interface
(c) Parameters (d) Functions

[Ans. (d) Functions]

2. _____ are the variables in a function definition.

- (a) Arguments (b) Parameters
(c) Identifiers (d) Operators

[Ans. (b) Parameters]

3. Explicitly _____ the types can help with debugging.

- (a) defining (b) annotating
(c) informing (d) computing

[Ans. (b) annotating]

4. All functions are _____ definitions.

- (a) static (b) dynamic
(c) algorithmic (d) static

[Ans. (a) static]

5. A _____ combines the external interface with an implementation of the interface.

- (a) parameter without type
(b) class declaration
(c) function definition
(d) parameter with type

[Ans. (b) class declaration]

6. In object oriented programs _____ are the interface

- (a) Implementation (b) parameters
(c) classes (d) Arguments

[Ans. (c) classes]

7. In object oriented programs, how the object is processed and executed is _____.

- (a) Implementation (b) Interface
(c) recursion (d) function

[Ans. (a) Implementation]

8. Stolen is an example _____ function.

- (a) user defined (b) impure
(c) pure (d) recursive

[Ans. (c) pure]

9. Evaluation of _____ functions does not cause any side effects to its output?

- (a) Impure (b) pure
(c) Recursive (d) built-in

[Ans. (b) pure]

CHOOSE THE CORRECT STATEMENT

1. (i) Algorithms are not expressed using statements of a programming language.

(ii) An interface is a set of action that an object can do

(iii) Implementation does not carries out the instructions defined in the interface.

(iv) Pure functions will give exact result.

- (a) i and iii (b) ii and iv
(c) iii and ii (d) i, ii and iv

[Ans. (b) ii and iv]

VERY SHORT ANSWERS

2 MARKS

1. Differentiate parameters and arguments.

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

2. Give an example of function definition parameter without type.

Ans. (requires: b>=0)

(returns: a to the power of b)

let rec pow a b:=

if b=0 then 1

else a * pow a (b-1)

3. Give an example of function definition parameter with type.

Ans. (requires: $b \geq 0$)
 (returns: a to the power of b)
 let rec pow (a: int) (b: int) : int :=
 if $b=0$ then 1
 else $a * \text{pow } a (b-1)$

4. What is recursive function?

Ans. A function definition which call itself is called recursive function.

5. Give an example of pure function.

Ans. let length s:=
 i := 0;
 if $i < \text{strlen}(s)$ then
 -- Do something which doesn't affect s
 ++i

6. Give an example of impure function.

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

7. Construct on algorithm that arranges meetings between these two types so that they change their color to the third type. In the end, all should display the same color.

Ans. let rec monochromatize a b c :=
 if $a > 0$ then
 a, b, c := a-1, b-1, c+2
 else
 monochromatize abc
 return a, b, c

SHORT ANSWERS

3 MARKS

1. Write an algorithm to check whether the entered number is even or odd.

Ans. (requires: $x \geq 0$)
 let rec even x :=
 $x=0 \parallel \text{odd}(x-1)$
 and odd x :=
 $x < 0 \ \&\& \text{even}(x-1)$

The value of the expression (T/F) is automatically the return value of the function, the last expression evaluates to a boolean and is returned automatically.

2. Explain the syntax of function definitions.

Ans. (i) The syntax to define functions is close to the mathematical usage: the definition is introduced by the keyword **let**, followed by the **name** of the function and its arguments; then the formula that computes the image of the argument is written after an **:=** sign. If you want to define a recursive function: use **"let rec" instead of "let"**.

(ii) Syntax : The syntax for function definitions:

let rec fn a1 a2 ... an := k

(iii) Here the '**fn**' is used as a function name. The names '**a1**' to '**an**' are variables used as parameters. The keyword '**rec**' is required if '**fn**' is to be a recursive function; otherwise it may be omitted.

3. Write a short note on syntax for function types.

Ans. The syntax for function types :

$x \rightarrow y$

$x1 \rightarrow x2 \rightarrow y$

$x1 \rightarrow \dots \rightarrow xn \rightarrow y$

The '**x**' and '**y**' are variables indicating types. The type $x \rightarrow y$ is the type of a function that gets an input of type '**x**' and returns an output of type '**y**'. Whereas $x1 \rightarrow x2 \rightarrow y$ is a type of a function that takes two inputs, the first input is of type '**x1**' and the second input of type '**x2**', and returns an output of type '**y**'. Likewise $x1 \rightarrow \dots \rightarrow xn \rightarrow y$ has type '**x**' as input of n arguments and '**y**' type as output.



Chapter-2

DATA ABSTRACTION

CHAPTER SNAPSHOT

- 2.1 Data Abstraction – Introduction
- 2.2 Abstract Data Types
- 2.3 Constructors and Selectors
- 2.4 Representation of Abstract datatype using Rational numbers
- 2.5 Lists, Tuples
 - 2.5.1 List
 - 2.5.2 Tuple
- 2.6 Data Abstraction in Structure

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

1. Which of the following functions that build the abstract data type ? *[Sep-2020; Aug-2021; July-'22; Mar-'25]*
 - (a) Constructors (b) Destructors
 - (c) Recursive (d) Nested

[Ans. (a) Constructors]
2. Which of the following functions that retrieve information from the data type? *[FRT & May-'22; QY-'23 & '24]*
 - (a) Constructors (b) Selectors
 - (c) Recursive (d) Nested

[Ans. (b) Selectors]
3. The data structure which is a mutable ordered sequence of elements is called
 - (a) Built in (b) List
 - (c) Tuple (d) Derived data

[Ans. (b) List]
4. A sequence of immutable objects is called *[Mar.-2020; June-'23, '24 & '25]*
 - (a) Built in (b) List
 - (c) Tuple (d) Derived data

[Ans. (c) Tuple]
5. The data type whose representation is known are called *[PTA-2; QY-2019; Mar-'23]*
 - (a) Built in datatype
 - (b) Derived datatype
 - (c) Concrete datatype
 - (d) Abstract datatype

[Ans. (c) Concrete datatype]
6. The data type whose representation is unknown are called
 - (a) Built in datatype (b) Derived datatype
 - (c) Concrete datatype
 - (d) Abstract datatype

[Ans. (d) Abstract datatype]
7. Which of the following is a compound structure?
 - (a) Pair (b) Triplet
 - (c) Single (d) Quadrat

[Ans. (a) Pair]
8. Bundling two values together into one can be considered as *[Govt. MQP - 2019; PTA-4; HY-'23]*
 - (a) Pair (b) Triplet
 - (c) Single (d) Quadrat

[Ans. (a) Pair]
9. Which of the following allow to name the various parts of a multi-item object? *[PTA-6; QY-'24]*
 - (a) Tuples (b) Lists
 - (c) Classes (d) Quadrats

[Ans. (c) Classes]

10. Which of the following is constructed by placing expressions within square brackets?

- (a) Tuples (b) Lists
(c) Classes (d) Quadrats

[Ans. (b) Lists]

PART - II

ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

1. What is abstract data type?

[Aug-2021; FRT & May-'22; Mar-'24]

Ans. Abstract Data Type (ADT) is a type for object whose behavior is defined by a set of values and operations.

2. Differentiate constructors and selectors.

Ans. [PTA-2, 3; QY-2019 & '23; July-'22; June-'24 & '25]

S. No.	Constructors	Selectors
(i)	Constructors are functions that build the abstract data type.	Selectors are functions that retrieve information from the data type.
(ii)	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.

[Mar.-2020 & '25; June-'23; QY-'24]

Ans. (i) Any way of bundling two values together into one can be considered as a Pair.

(ii) **Example :** Lists

lst = [(0,10), (1,20)]

4. What is a List? Give an example. [QY - 2019]

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

Example : lst := [10, 20]

x, y := lst

5. What is a Tuple? Give an example. [Mar-'23]

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

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

PART - III

ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. Differentiate Concrete datatype and Abstract datatype. [HY-'23; QY-'24]

Ans.

S. No.	Concrete datatype	Abstract datatype
(i)	Concrete Data Types or structures (CDT's) are direct implementations of a relatively simple concept.	Abstract Data type (ADT) is a type for objects whose behavior is defined by a set of values and operations.
(ii)	A concrete data type is a data type whose representation is known.	Abstract data type the representation of a data type is unknown.

2. Which strategy is used for program designing? Define that Strategy. [Govt. MQP-2019]

Ans. A powerful strategy for designing programs: 'wishful thinking'. Wishful Thinking is the formation of beliefs and making decisions according to what might be pleasing to imagine instead of by appealing to reality.

3. Identify Which of the following are constructors and selectors? [PTA-5; FRT-22]

- (a) N1:=number()
(b) accetnum(n1)
(c) displaynum(n1)
(d) eval(a/b)
(e) x,y:= makeslope (m), makeslope(n)
(f) display()

Ans. (a) Constructor (b) Selector
(c) Selector (d) Selector
(e) Constructor (f) Selector

4. What are the different ways to access the elements of a list. Give example. [Mar-'24]

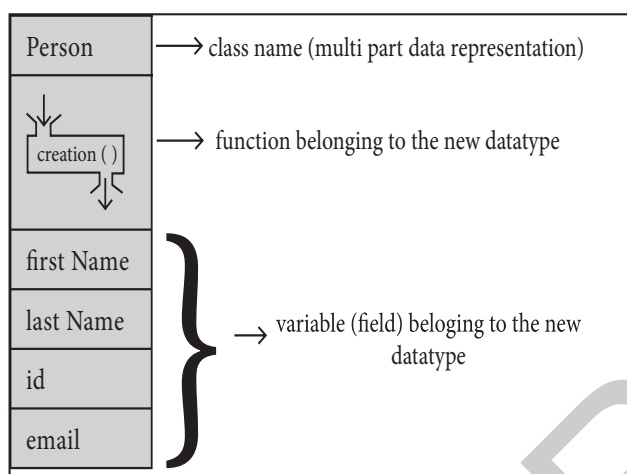
Ans. The elements of a list can be accessed in two ways.

- (1) **Multiple assignment:** This method which unpacks a list into its elements and binds each element to a different name.

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



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

- (ii) The class (structure) construct defines the form for multi-part objects that represent a person.
- (iii) Person is referred to as a class or a type, while p1 is referred to as an object or an instance.
- (iv) Here, class Person as a cookie cutter, and p1 as a particular cookie. Using the cookie

cutter you can make many cookies. Same way using class created many objects of that type.

- (v) A class defines a data abstraction by grouping related data items. A class is not just data, it has functions defined within it. We say such functions are subordinate to the class because their job is to do things with the data of the class.

PTA QUESTIONS AND ANSWERS

1 MARK

1. Expansion of ADT : [PTA-1]

- (a) Abstract Data Tuple
- (b) All Data Template
- (c) Abstract Data Type
- (d) Application Data Type

[Ans. (c) Abstract Data Type]

2. ADT can be implemented using _____. [PTA-5]

- (a) singly linked list
- (b) doubly linked list
- (c) either A or B
- (d) neither A nor B

[Ans. (c) either A or B]

GOVERNMENT EXAM QUESTIONS AND ANSWERS

1 MARK

1. The datatype whose representation is unknown is called [HY-2019; Mar-'24]

- (a) Built-in datatype
- (b) Derived datatype
- (c) Concrete datatype
- (d) Abstract datatype

[Ans. (d) Abstract datatype]

2 MARKS

1. Define Abstraction. [Aug-2021]

Ans. The process of providing only the essentials and hiding the details is known as abstraction.

3 MARKS

1. (a) What is selector? [Sep-2020]
(b) What are the parts of a program?

Ans. (a) Selectors are the functions that retrieve information from the data type.

City: = make city (name, lat, lon)
Creates a city object and

- (i) getname(city)
- (ii) getlat(city)
- (iii) getlon(city)

- are the selectors because these functions extract the information of the city object
- (b) The two parts of a program are, the part that operates on abstract data and the part that defines a concrete representation, is connected by a small set of functions that implement abstract data in terms of the concrete representation.

ADDITIONAL QUESTIONS AND ANSWERS

CHOOSE THE CORRECT ANSWER **1 MARK**

- Which of the following is a powerful concept that allows programmers to treat code as objects?
(a) Encapsulation (b) Data Abstraction
(c) Inheritance (d) Polymorphism
[Ans. (b) Data Abstraction]
- Which of the following provides modularity?
(a) Datatypes (b) Subroutines
(c) Classes (d) Abstraction
[Ans. (d) Abstraction]
- Which of the following is a type for objects whose behavior is defined by a set of value and a set of operations?
(a) User-defined datatype
(b) Derived datatype
(c) Built-in datatype (d) Abstract datatype
[Ans. (d) Abstract datatype]
- ADT behavior is defined by
(i) Set of Variables (ii) Set of Value
(iii) Set of Functions (iv) Set of Operations
(a) i, ii (b) ii, iii
(c) ii, iv (d) i, iii
[Ans. (c) ii, iv]
- The process of providing only the essentials and hiding the details is known as
(a) Functions (b) Abstraction
(c) Encapsulation (d) Pairs
[Ans. (b) Abstraction]
- Which of the following gives an implementation independent view?
(a) Abstract (b) Concrete
(c) Datatype
(d) Behavior of an object [Ans. (a) Abstract]

- How many ways to implement an ADT?
(a) Only one (b) Two
(c) Three (d) Many
[Ans. (b) two]
- Which of the following are implemented using lists?
(a) Singly linked list ADT
(b) Doubly Linked list ADT
(c) Stack ADT (d) Queue ADT
(e) All of these [Ans. (e) All of these]
- Which of the following replicate how we think about the world?
(a) Queue ADT (b) Data Hiding
(c) Data Abstraction (d) Stack ADT
[Ans. (c) Data Abstraction]
- To facilitate data abstraction, How many types of functions are created?
(a) 2 (b) 3
(c) 4 (d) Only one
[Ans. (a) 2]
- Which of the following function that facilitate the data abstraction?
(a) Constructors (b) Destructors
(c) Selectors (d) a and c
[Ans. (d) a and c]
- Which of the following are functions that build the abstract datatype?
(a) Constructors (b) Destructors
(c) Selectors (d) All of these
[Ans. (a) Constructors]
- Which of the following extract the information of the object?
(a) Constructors (b) Functions
(c) Selectors (d) Destructors
[Ans. (c) Selectors]
- In which data representation, a definition for each function is known.
(a) User defined (b) Built-in
(c) Abstract (d) Concrete
[Ans. (d) Concrete]
- How many parts are there in the program?
(a) 2 (b) 3
(c) 4 (d) Many
[Ans. (a) 2]
- How many values can be stored in the list?
(a) 4 (b) 10
(c) 100 (d) Multiple
[Ans. (d) Multiple]

17. To implement the concrete level of data abstraction the language python provides a compound structure called

- (a) ADT (b) Concrete data
(c) Pair
(d) User defined function [Ans. (c) Pair]

18. Which of the following is constructed by placing expressions within square brackets separated by commas?

- (a) List (b) Tuple
(c) Set (d) Dictionary
[Ans. (a) List]

19. `l = [10, 20]` is an example

- (a) Tuple (b) Set
(c) List (d) Dictionary
[Ans. (c) List]

20. List can also be called as

- (a) Functions (b) Class
(c) Structure (d) Pairs
[Ans. (d) Pairs]

21. How many ways are there to represent pair datatype?

- (a) 2 (b) 4 (c) 3 (d) 5
[Ans. (a) 2]

22. `Color = ('red', 'green', 'blue')` is an example of

- (a) Dictionary (b) List
(c) Set (d) Tuple
[Ans. (d) Tuple]

23. Which of the following does not allow us to name the various parts of a multi-item object?

- (a) List (b) Tuple
(c) Pair (d) All of these
[Ans. (d) All of these]

24. Which of the following defines a data abstraction by grouping related data items?

- (a) List (b) Pair
(c) Class (d) Tuple
[Ans. (c) Class]

25. Which of the following as bundled data and the functions that work on that data?

- (a) Object (b) Pair
(c) List (d) Class
[Ans. (d) Class]

26. CDT expansion is

- (a) Collective Data Type (b) Class Data Type
(c) Concrete Data Type
(d) Central Data Type
[Ans. (c) Concrete Data Type]

MATCH THE FOLLOWING

	List I		List II
1.			
i)	List	1)	arr (1,2,3,4)
ii)	Tuples	2)	getname (city)
iii)	Class	3)	Student [rno, name, mark]
iv)	Selectors	4)	arr [1,2,3,4]

- (i) (ii) (iii) (iv)
(a) 1 2 3 4
(b) 4 3 2 1
(c) 4 1 3 2
(d) 3 2 4 1

[Ans. (c) (i)-4; (ii)-1; (iii)-3; (iv)-2]

CHOOSE THE ODD MAN OUT

1. (a) List
(b) Multiple assignment
(c) Classes
(d) Element selection operator
[Ans. (c) Classes]

CHOOSE AND FILL IN THE BLANKS

1. Data Abstraction allows programmers to treat code as _____.
(a) Objects (b) Classes
(c) Members (d) Parameters
[Ans. (a) Objects]

2. _____ are the representation for Abstract Data types.
(a) Objects (b) Classes
(c) Functions (d) Lists
[Ans. (b) Classes]

3. Classes are the representation for _____.
(a) Abstract datatype
(b) Built-in datatype
(c) Concrete datatype
(d) Essential datatype
[Ans. (a) Abstract datatype]

4. The _____ can be implemented using singly linked list or doubly linked list.
(a) Tuple ADT (b) List ADT
(c) Function ADT (d) List ADT
[Ans. (b) List ADT]

5. The basic idea of _____ is to structure programs so that they operate on abstract data
(a) Encapsulation (b) Polymorphism
(c) Data type (d) Data Abstraction
[Ans. (d) Data Abstraction]

6. A _____ data representation is defined as an independent part of the program.

- (a) Abstract (b) Concrete
(c) List (d) Tuple

[Ans. (b) Concrete]

7. _____ are functions that retrieve information from the data type.

- (a) Constructors (b) Selectors
(c) List (d) Tuples

[Ans. (b) Selectors]

8. _____ is made up of list or Tuples.

- (a) Set (b) Pair
(c) Dictionary
(d) Control Structures

[Ans. (b) Pair]

9. List is constructed by using _____ and _____.

- (a) (), , (b) <>, ;
(c) [], , (d) [], :

[Ans. (c) [], ,]

10. A _____ is a comma separated values surround with parentheses.

- (a) List (b) Tuple
(c) Set (d) Dictionary

[Ans. (b) Tuple]

11. Tuple is constructed by using _____ and _____.

- (a) (), (b) [], (c) [],: (d) (),:

[Ans. (a) (),]

12. A _____ is not just data, it has functions defined within it.

- (a) Class (b) List
(c) Pair (d) Object

[Ans. (a) Class]

CHOOSE THE INCORRECT STATEMENT

- (i) ADT is defined by set of values and set of operations
(ii) ADT does specify how data will be organized in the memory.
(iii) Constructors are not used to built abstract data type.
(iv) Selectors are functions that retrieve information from the data type.
(a) i, ii (b) i, iv
(c) ii, iii (d) i, iii, iv

[Ans. (b) i, iv]

CHOOSE THE INCORRECT PAIR

- (a) Abstraction – hiding the details
(b) Abstract data type–constructor & destructor
(c) Abstraction – providing only the essentials
(d) Abstract data type – constructor & selectors
[Ans. (b) Abstract data type – Constructor & destructor]

VERY SHORT ANSWERS

2 MARKS

1. Give an example of implementing an ADT.

- Ans. (i)** There can be different ways to implement an ADT, for example, the List ADT can be implemented using singly linked list or doubly linked list.
(ii) Similarly, stack ADT and Queue ADT can be implemented using lists.

2. Identify which is the constructor and selector from the following statement.

- (i) The Functions that retrieve information from the datatype
(ii) The function which creates an object.

- Ans. (i)** Selector
(ii) Constructor

3. Write the pseudo code for the representation of the rational number.

- Ans.** The pseudo code for the representation of the rational number
x,y:=8,3
rational(x,y)
numer(x)/denom(y)
- - output : 2.6666666666666665

4. How the data abstraction implemented?

- Ans. (i)** To implement the data abstraction programming languages like Python provides a compound structure called Pair which is made up of list or Tuple.
(ii) The first way to implement pairs is with the List construct.

5. Write a pseudocode to represent rational numbers using list.

- Ans.** rational(n, d):
 return [n, d]
numer(x):
 return x[0]
denom(x):
 return x[1]

6. How a class defines a data abstraction?

- Ans. (i)** A class defines a data abstraction by grouping related data items. A class is not just data, it has functions defined within it.
- (ii)** Functions are subordinate to the class because their job is to do things with the data of the class.

7. From the statement $P1 := \text{Person}()$, What does P1 and person referred.

- Ans.** Person is referred to as a class or a type, while p1 is referred to as an object or an instance.

8. How the elements of a list can be accessed?

- Ans. (i)** The elements of a list can be accessed in two ways.
- (ii)** The first way is via multiple assignment and the second method is by the element selection operator.

SHORT ANSWERS**3 MARKS****1. Identify the constructor and selector from the following.**

- (i)** City = Make city (name, lat, lon)
(ii) Get name (city)
(iii) Make point (x,y)
(iv) x coord (point)
(v) y coord (point)

- Ans. (i)** Constructor **(ii)** Selector
(iii) Constructor **(iv)** Selector
(v) Selector

2. Write a note on Data Abstraction.

- Ans. (i)** Data abstraction is used to define an Abstract Data Type (ADT), which is a collection of constructors and selectors.
- (ii)** Constructors create an object, bundling together different pieces of information, while selectors extract individual pieces of information from the object.

3. Give an example of an ADT for rational numbers.

- Ans. An ADT for rational numbers :**

- - constructor
- - constructs a rational number with numerator x, denominator y

rational(x, y)

- - selector

numer(x) → returns the numerator of rational number x

denom(y) → returns the denominator of rational number y

LONG ANSWERS**5 MARKS****1. Explain the representation of Abstract datatype using rational numbers.**

- Ans. (i)** The basic idea of data abstraction is to structure programs so that they operate on abstract data.

(ii) At the same time, a concrete data representation is defined as an independent part of the program.

(iii) Any program consist of two parts. The two parts of a program are, the part that operates on abstract data and the part that defines a concrete representation, is connected by a small set of functions that implement abstract data in terms of the concrete representation.

(iv) Example : Rational numbers

A rational number is a fraction of integers ie., numerator/denominator.

For example 8/3 or 19/23

(v) Constructor builds a rational number rational (x, y)

(vi) Selector get parts of the rational number.
 numer (x)
 denom (y)

(vii) Pseudo code representation of rational number.

x,y:=8,3

rational(x,y)

numer(x)/denom(y)

- - output : 2.6666666666666665

