aglasem.com

FOR GOA CLASS 12 PREPARATION

# Goa Board Class 12 Sample Paper 2027 Software Technology

| EXAM | YEAR | TYPE |
| --- | --- | --- |
| Goa Class 12 | 2027 | Sample Paper |

# GOA BOARD OF SECONDARY & HIGHER SECONDARY EDUCATION

ALTO – BETIM, GOA

## PROFORMA FOR ANALYSIS OF MODEL QUESTION PAPER 2026-2027

**Subject: Software Technology**
**Duration: 2 hrs** **Class: XII Computer Technique** **Marks Allotted: 50**

**1. Sections if any: - YES – Section A, B, C and D.**

**2. Total No. of questions and marks: -**

| No. of questions set in the paper | No. of questions to be attempted | Total marks for all the questions set in the paper | Maximum marks for attempted questions |
|---|---|---|---|
| 26 | 26 | 50 | 50 |

**3. Study of options:**

(a) Overall options in the whole question paper:- -NO-
(b) Overall options within the sections: - -NO-
(c) Overall options within a question: - -NO-
(d) Internal option within the question: - -YES-
(e) Comparability options: --NO-

**4. Instructions for examiners: -** Evaluate as per pattern

**5. Directions in individual question: - -**NIL

**6. Allocation of marks question-wise and part-wise: -**

Section A: Q.Nos.1 to 10 – 1 mark each
Section B: Q.Nos. 11 to 20 – 2 marks each
Section C: Q.Nos. 21 to 24 – 3 marks each
Section D: Q.Nos. 25 to 26 – 4 marks each

**7. General format**

(a) Arrangement of question, numbering and printing etc.

| Section A | Section B | Section C | Section D |
|---|---|---|---|
| 1 | 11 | 21 | 25 |
| 2 | 12 | 22 | 26 |
| 3 | 13 | 23 | |

| | | | |
|---|---|---|---|
| 4 | 14 | 24 | |
| 5 | 15 | | |
| 6 | 16 | | |
| 7 | 17 | | |
| 8 | 18 | | |
| 9 | 19 | | |
| 10 | 20 | | |

**8. Question-wise Analysis**

| Sr. no. Part no. | Objective | Content Area Major area of content topic | Form of the questions | Language of Questions | Definiteness of the answers | Difficulty level of the questions | Marks allotted | Time in min. |
|---|---|---|---|---|---|---|---|---|
| 1 | K | Review of Python Programming | MCQ | ENGLISH | Definite | Easy | 1 | 2 |
| 2 | K | | MCQ | | Definite | Easy | 1 | 2 |
| 3 | A | | MCQ | | Definite | Average | 1 | 2 |
| 4 | U | | MCQ | | Definite | Average | 1 | 2 |
| 5 | U | Networking | MCQ | | Definite | Average | 1 | 2 |
| 6 | U | | MCQ | | Definite | Easy | 1 | 2 |
| 7 | U | | MCQ | | Definite | Easy | 1 | 2 |
| 8 | U | | MCQ | | Definite | Average | 1 | 2 |
| 9 | U | Dictionary | MCQ | | Definite | Easy | 1 | 2 |
| 10 | K | Inheritance | MCQ | | Definite | Easy | 1 | 2 |
| 11 | K | Dictionary | SA | | Definite | Average | 2 | 4 |
| 12 | U | Networking | SA1 | | Definite | Average | 2 | 4 |
| 13 | U | | SA1 | | Definite | Average | 2 | 4 |
| 14 | K | Tuples | SA1 | | Definite | Average | 2 | 4 |
| 15 | U | Basic Concepts of OOPs | SA1 | | Definite | Average | 2 | 4 |
| 16 | K | | SA1 | | Definite | Easy | 2 | 4 |
| 17 | K | | SA1 | | Definite | Easy | 2 | 4 |
| 18 | K | Classes in Python | SA1 | | Definite | Easy | 2 | 4 |
| 19 | U | | SA1 | | Definite | Average | 2 | 4 |
| 20 | K | Inheritance | SA1 | | Definite | Easy | 2 | 4 |
| 21 | A | Networking | SA2 | | Definite | Average | 3 | 9 |
| 22 | A | | SA2 | | Definite | Average | 3 | 9 |
| 23 | U | Basic Concepts of OOPs | SA2 | | Definite | Average | 3 | 9 |
| 24 | A | Dictionary | SA2 | | Partially Definite | Average | 3 | 9 |
| 25 | S | Dictionary | LA | | Partially Definite | Difficult | 4 | 12 |
| 26 | S | Tuples | LA | | | Difficult | 4 | 12 |

**A) Weightage to Learning Outcomes**

| Sr. No. | Learning Outcomes | Marks | Percentage of Marks |
|---|---|---|---|
| 1 | Knowledge | 13 | 26% |
| 2 | Understanding | 17 | 34% |
| 3 | Application | 12 | 24% |
| 4 | Skill | 8 | 16% |

**B) Weightage to Content/Subject Units**

| Sr. No. | Units | No. of Questions | Marks | Percentage |
|---|---|---|---|---|
| 1 | Review of Python Programming | 4 | 4 | 8% |
| 2 | Dictionaries | 4 | 10 | 20% |
| 3 | Tuples | 2 | 6 | 12% |
| 4 | Basic Concepts of OOPs | 4 | 9 | 18% |
| 5 | Classes | 2 | 4 | 8% |
| 6 | Inheritance | 2 | 3 | 6% |
| 7 | Networking | 8 | 14 | 28% |

**C) Coverage of topics**

| No. of topics in the syllabus | Topics covered | Percentage |
|---|---|---|
| 7 | 7 | 100% |

**D) Weightage to form of questions**

| Sr. No. | Form of questions | Total Marks | Percentage |
|---|---|---|---|
| 1 | Multiple Choice Questions (MCQ) | 10 | 20% |
| 2 | Short Answer Type (SA1) | 20 | 40% |
| 3 | Short Answer Type (SA2) | 12 | 24% |
| 4 | Long Answer Type(LA) | 8 | 16% |
| | Total | | 100% |

The expected time for different types of question would be as follows:

| Sr. No. | Form of questions | Approx. Time for each question in mins(t) | No. of questions (n) | Approx. Time for each form of question in mins (n x t) |
|---|---|---|---|---|
| 1 | Multiple Choice Questions (MCQ) | 2 | 10 | 20 |
| 2 | Short Answer Type (SA1) | 4 | 10 | 40 |
| 3 | Short Answer Type (SA2) | 9 | 4 | 36 |
| 4 | Long Answer Type(LA) | 12 | 2 | 24 |
| | Total | | 26 | 120 |

As the total time is calculated on the basis of the number of questions required to be answered and the length of their anticipated answers, it would, therefore be advisable for the candidates to budget their time properly by cutting out the superfluous words and be within the expected time limits.

**Scheme of Options**

There will be no overall choice. However, there is an internal choice in Section C, Q.No.24 & section D, Q.No.26.

**E) Definiteness of Questions:**

| Sr. No. | Definiteness | No. of questions | Marks | Percentage |
|---|---|---|---|---|
| 1 | Definite | 23 | 39 | 78% |
| 2 | Partly Definite | 3 | 11 | 22% |
| 3 | Not Definite | Nil | -- | -- |

**F) Weigh Difficulty level of questions:**

| Sr. No. | Level of difficulty | No. of question | Marks | Percentage |
|---|---|---|---|---|
| 1. | Easy | 10 | 14 | 28% |
| 2. | Average | 14 | 28 | 56% |
| 3. | Difficult | 2 | 8 | 16% |

A question may vary in difficulty level from individual to individual. As such, the assessment in respect of each question will be made by the paper setter on the basis of general anticipation from the group as a whole taking the examination. This provision is only to make the paper balanced in its weightage, rather than to determine the pattern of marking at any stage.

# Goa Board of Secondary & Higher Secondary Education, Alto, Betim - Goa

## BLUE PRINT

**Subject : Software Technology** **Timing : 2 hours** **Std : XII Computer Technique**

| | **Objective** | **Knowledge** | | | | **Understanding** | | | | **Application** | | | | **Skill** | | | | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S.N | Contents Units | MCQ (1) | SA-1 (2) | SA-II (3) | LA (4) | MCQ (1) | SA-1 (2) | SA-II (3) | LA (4) | MCQ (1) | SA-I (2) | SA-II (3) | LA (4) | MCQ (1) | SA-I (2) | SA-II (3) | LA (4) | **Total (50)** |
| 1 | Review of Python Programming | 2(1) | | | | 1(1) | | | | 1(1) | | | | | | | | **4(4)** |
| 2 | Dictionaries | 1(1) | 1(2) | | | | | | | | | 1(3)* | | | | | 1(4) | **4(10)** |
| 3 | Tuples | | | | | | | | | | 1(2) | | | | | | 1(4)* | **2(6)** |
| 4 | Basic Concepts of OOPs | | 2(2) | | | | 1(2) | 1(3) | | | | | | | | | | **4(9)** |
| 5 | Classes | | 1(2) | | | | 1(2) | | | | | | | | | | | **2(4)** |
| 6 | Inheritance | | 1(2) | | | 1(1) | | | | | | | | | | | | **2(3)** |
| 7 | Networking | | | | | 4(1) | 2(2) | | | | | 2(3) | | | | | | **8(14)** |
| | **Total** | **8(13)** | | | | **11(17)** | | | | **5(12)** | | | | **2(8)** | | | | **26(50)** |

Note: Figures outside the brackets indicate the no. of questions and those inside the brackets indicate the marks.

# SYLLABUS

## GOA BOARD OF SECONDARY AND HIGHER SECONDARY EDUCATION

**ALTO BETIM – GOA 403 521**

## Course: Computer Technique

**SUBJECT: SOFTWARE TECHNOLOGY** **Std: XII**

**THEORY – 50 marks**

| Chapter No | Syllabus | marks |
|---|---|---|
| 1 | **Review of Python Programming**<br>Review of Python programming concepts taught in Std.XI (Theory content of if else, WHILE loop, FOR loop, Functions, Lists and strings to be asked for the exam.) | 4 |
| 2 | **Dictionaries**<br>2.1 What is dictionary<br>2.2 Creation, initializing dictionaries<br>2.3 Accessing the elements in a Dictionary<br>2.4 Traversing a dictionary<br>2.5 Dynamic creation of dictionary<br>2.6 Appending values to the dictionary<br>2.7 Merging dictionaries<br>2.8 Removing an item from dictionary<br>2.9 Dictionary functions and methods<br>2.9.1 cmp()<br>2.9.2 len()<br>2.9.3 clear()<br>2.9.4 get(key,value)<br>2.9.5 has_key(key)<br>2.9.6 items()<br>2.9.7 keys()<br>2.9.8 values() | 10 |
| 3 | **Tuples**<br>3.1 What is a Tuple<br>3.2 Tuple Creation<br>3.3 Add new element to Tuple<br>3.4 Tuple Assignment<br>3.5 Tuple Functions<br>3.5.1 sorted()<br>3.5.2 len()<br>3.5.3 max()<br>3.5.4 min()<br>3.5.5 tuple() | 6 |

| | | |
|---|---|---|
| 4 | **Basic concepts of OOPS**<br>4.1 Introduction to OOP in Python<br>4.2Difference between OOP and Procedural oriented programming<br>4.3 What are classes and objects<br>4.4 What is OOP methodology<br>4.5 Definition of Encapsulation<br>4.6 Definition of Data Abstraction<br>4.7 Definition of Inheritance<br>4.8 Definition of Polymorphism<br>4.8.1 Operator Overloading<br>4.8.2 Function Overloading<br>4.9 Advantages of OOPS | 9 |
| 5 | **Classes in Python**<br>5.1 Name spaces and scope rules<br>5.2 Defining Classes and Objects,<br>5.3 Class Attributes v/s Instance Attributes | 4 |
| 6 | **Inheritance**<br>6.1 Concept of Inheritance<br>6.2 Types of Inheritance<br>6.3 Definition of Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance and Hybrid Inheritance. | 3 |
| 7 | **Networking**<br>7.1 Computer Networks<br>7.1.1 What is a Computer Network<br>7.1.2 Advantages, characteristics of Network<br>7.2. Types of Networks<br>7.2.1 LAN<br>7.2.2 MAN<br>7.2.3 WAN<br>7.2.4 Wired Network<br>7.2.5 Wireless Network<br>7.3. Data Transmission<br>7.3.1 What is data transmission<br>7.3.2 Data Communication<br>7.3.3 Modes of data transmission<br>7.3.3.1 Simplex<br>7.3.3.2 Duplex<br>7.3.3.3 Full Duplex<br>7.3.4 Types of Signals<br>7.3.4.1 Analog<br>7.3.4.2 Digital<br>7.4 Physical Topologies<br>7.4.1 Linear Bus | 14 |

| | | |
|---|---|---|
| | 7.4.2 Ring<br>7.4.3 Star<br>7.4.4 Tree<br>7.4.5 Mesh Topology<br>7.5 Client server architecture<br>7.6 Peer to peer | |

**Note: As of 22/04/2026 Chapter No.2 – Dictionary, 2.9.1 cmp( ) and 2.9.5 has_key( ) is no longer supported by python in the latest version hence it is not included in the above topic.**

**Practicals**

List of practicals to be completed:

For the 1st Term 10 practicals using Dictionaries.

For the 2nd Term 10 practicals using Tuples.

Total 20 practicals need to be kept in the file.

**Using Dictionaries**

1. To input 'n' names and phone numbers to store it in the dictionary and to input any
name and to print the phone number of that particular name.
2. To input 'n' employee number and name and to display all employee's information in
ascending order based upon their number.
3. To create a phone book and delete a particular number using the name.
4. To input any 5 years and the population of any city and print it on the screen.

(6 more programs using dictionaries to be included)

**Using Tuples**

1. To input 5 subject names and put it in a tuple and display the tuple information on the output screen.
2. To input any two tuples and interchange the tuple values.
3. To input 'n' employee's salary and find the minimum and maximum salary among 'n' employees.
4. To input 'n' numbers and separate the tuple in the following manner.
   Example: T= (10,20,30,40,50,60)
   T1= (10,30,50)
   T2= (20,40,60)
5. Write a Python program to calculate the product by multiplying all the numbers of a given tuple.

(5 more programs using tuples to be included)

**Marks distribution for the practical exams will be as follows:**

1. File - File must contain all practicals: 10 marks
2. Writeup - Written program (to be kept as record: 10 marks
3. Viva - Oral questions from the corresponding topic: 10 marks
4. Debugging/Innovative/Logic used/Comments - Marks for doing it differently and on time. : 10 marks
5. Input/Output - Accuracy, validations used: 10 marks

# MODEL QUESTION PAPER 2026-2027

**Time: 2 Hours** **SOFTWARE TECHNOLOGY**

**Subject Code**

| V | 5 | 3 | 1 | 5 |
|---|---|---|---|---|

**Total No. Of Questions:26** **Maximum Marks:50**

Instructions : i. All questions are compulsory
ii. There are four sections in this question paper (A,B,C & D)
iii. Section A there are ten MCQ of 1 mark each.
iv. Section B contains ten questions of 2 marks each.
v. Section C contains four questions of 3 marks each. Internal choice is provided for Q24.
vi. Section D contains two questions of 4 marks each. Internal choice is provided for Q26.
vii. Write the number of each question clearly on the answer book .

## Section A

**Question numbers from 1 to 10 carry 1 mark each.**

1. When does a while loop in python stop executing?
   - When the loop condition is true
   - When the loop condition becomes false.
   - Only when continue statement is encountered.
   - After the third iteration.

2. What is the index of the rightmost character of a string while traversing from right to left.
   - 0
   - 1
   - -1
   - 2

3. Given a list L=[10,20,30,40,50,60,70], what would L[1:4] return?
   - [10,20,30,40]
   - [20,30,40,50]
   - [20,30,40]
   - [30,40,50]

4. Which of the following keywords is used for function declaration in Python language?

- def
- function_name
- define
- None of the these

5. Which of the following transmission directions listed is not a legitimate channel?

- Simplex
- Half Duplex
- Duplex
- Double Duplex

6. The type of network that spans the entire world is:

- Wide Area Network
- Private Network
- Metropolitan Network
- Local Area Network

7. In which network topology the data is passed through a central concentrator or hub or a switch.

- Linear Bus topology
- Ring Topology
- Mesh Topology
- Star Topology

8. Which network uses radio waves to communicate with other system

- Wireless Network
- Wired Network
- WAN
- MAN

9. Which of the following function will return the key value pair of a dictionary?

- keys()
- values()
- items()
- all of the above

10. Another term for derived class in inheritance?

- parent
- multilevel
- base
- child

## Section B

**Question numbers from 11 to 20 carry 2 marks each.**

11. Explain with an example the keys() built-in Dictionary method.

12. Explain half-duplex mode of Data transmission.

13. Explain Analog and Digital Signals.

14. Write the syntax to create a tuple using built in function and add new elements to it.

15. What is Data Encapsulation in Object Oriented Programming?

16. Write two differences between object oriented programming and Procedural oriented Programming.

17. What is Data Abstraction in Object Oriented Programming?

18. Why do we use classes in Python?

19. What is meant by class attributes in Python?

20. Why is inheritance an important aspect of the object-oriented paradigm?

## Section C

**Question numbers from 21 to 24 carry 3 marks each.**

21. With a neat labeled diagram, describe Linear Bus topology.

22. What is a Computer Network? State two advantages of a Computer Network.

23. What is Polymorphism? Explain operator overloading and function overloading.

24. Write a python code to add 'n' book ID and their prices to a dictionary and print the dictionary. Display the ID of books with price above 250.

**OR**

24. Explain with example how two dictionaries can be merged into one dictionary.

## Section D

**Question numbers from 25 to 26 carry 4 marks each.**

25. There are 'n' number of students in a class of Std.XI Science. Write a python program to enter the following details of the students using dictionaries.

    Name, Roll No

    Also input any Name of the student from the class and print the Roll No. pertaining to that particular student name.

26. Write a Python program to input any two tuples and interchange the tuple values.

**OR**

26. Write a python program to input n numbers in a tuple, then add a new element to that tuple and find the maximum and minimum value among the given numbers in the tuple.

**********

# Answer Key

## Section A

1. b) When the loop condition becomes false.
2. c) -1
3. c) [20,30,40]
4. a) def
5. d) Double Duplex
6. a) Wide Area Network
7. d) Star Topology
8. a) Wireless Network
9. c) items()
10. d) child

## Section B

11. keys(): It returns a list of key values in a dictionary, which is not in any particular order.

Syntax: D.keys()  # D is a dictionary

Example: D={'A':'Ant','B':'Bat','C':'Cat','D':'Dog'}

```
D.keys()
['A','B','C','D']
```

12. Half Duplex: Half-duplex data transmission means that data can be transmitted in both directions on a signal carrier, but not at the same time.

Example of half duplex is a walkie- talkie.

13. Analog Signals: Analog signals are continuous wave form and changes smoothly over time. These are represented by sine waves. Analog signals contain noise and wave distortions. Analog signals cannot perform high quality transmission.

Digital signals: **:** Digital signals are not continuous but are discrete in value and time. These signals are in the form of binary digits (i.e. 0,1). These are represented by square

waves. Noise and wave distortions are not found in Digital signals. Digital signals can perform high quality transmission. Data transmission between computers is in the form of digital signals.

14. Syntax:

```
t = tuple()   # t is a variable name and tuple is the built-in-function
We can add new element to tuple using + operator.
t = t + (60,)
```

Example:

```
t=tuple()
t=10,20,30
t=t+(60,)
print(t)
```

Output:

```
(10,20,30,60)
```

15. Data Encapsulation: It refers to the bundling of data with the methods that operate on the data. The data members can be accessed through its member functions only. It keeps data safe from any external interference.

16.

| Object Oriented Programming | Procedural Oriented Programming |
|---|---|
| 1. Object-oriented programming is an approach to problem-solving where all computations are carried out using objects. It follows a bottom-up approach | 1. Procedural programming uses a list of instructions to do computation step by step. It follows a top-down approach. |
| 2. The program is divided into objects | 2. Program is divided into functions or modules |
| 3. It provides data hiding so it is more secure | 3. It is less secure |
| 4. It supports inheritance | 4. It does not support inheritance |
| 5. Makes use of Access modifiers 'public', private', protected' | 5. Access modifiers are not supported. |

17. Data Abstraction: Data Abstraction is a process of hiding the internal details from the end user. Only the essential information is provided to the outside world while the internal details are hidden. For example a mobile phone user does not know how calls are made instead it provides an interface that hides these details from the user.

18. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state.

19. Class Attributes: Class Attributes are the variables defined directly in the class that are shared by all objects of the class. They are accessed using object dot notation.
    Eg. object.class_attribute

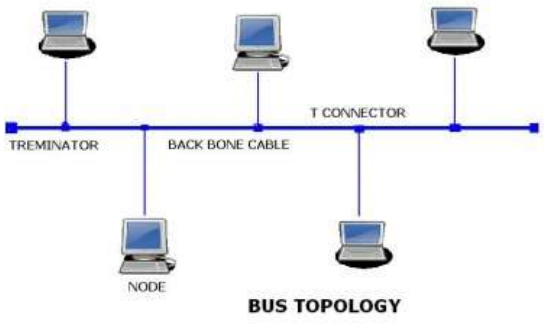
20. Inheritance is the process of forming new class from an existing class. The existing class is called base class and the new class created is called derived class or inherited class. The data and methods of the base class are accessible to the inherited class.

## Section C

21. Linear Bus/ Bus Topology: In a Bus Topology, all the nodes (File Servers, Workstations and peripherals) are connected by one single cable. A bus topology consists of a main run of cable with a terminator at each end. All nodes are connected to this linear cable. They are popular on LANs because they are inexpensive and easy to install.
    Advantages of Bus Topology:
    - Less expensive, easy to handle and implement.
    - Requires less cable.
    - It is best suited for small networks.

    Disadvantages of Bus Topology:
    - Since the cable length is limited, this limits the number of stations that can be connected.
    - Entire network fails if there is a break in the main cable.
    - Difficult to identify the problem if the entire network fails.

22. A computer network is a collection of computers connected to each other for the purpose of sharing information and resources. It allows the computers to communicate with each other and share resources which include information, software and peripheral devices such as printers, scanners etc.



BUS TOPOLOGY

    Advantages of a Computer Network:
    1. **File Sharing**: The major advantage of a computer network is that it allows file sharing and remote file access. A person sitting at one workstation that is connected to a network can easily send or receive files present on another

workstation, provided he is authorized to do so. This saves him/her the hassle of carrying a storage device every time data needs to be transported from one system to another.

2. **Resource Sharing:** Resource sharing is another important benefit of a computer network. For example, if there are twelve employees in an organization, each having their own computer, they will require twelve modems and twelve printers if they want to use the resources at the same time. A computer network, on the other hand, provides a cheaper alternative by the provision of resource sharing. All the computers can be interconnected using a network, and just one modem and printer can efficiently provide the services to all twelve users.
3. **Sharing Storage:** On a network, one can access data from any machine. Hence, storage can be distributed and thus database load can be shared on the network. A file can have copies on two or more machines .
4. **Cost effective:** Shared resources mean reduction in hardware costs. Shared files mean reduction in memory requirement, which indirectly means reduction in file storage expenses. A particular software can be installed only once on the server and made available across all connected computers at once. This saves the expense of buying and installing the same software as many times for as many users.
5. **Flexible:** A user can log on to a computer anywhere on the network and access his files. This offers flexibility to the user. A network also allows the network administrator to choose which user on the network has what specific permissions to handle a file.
6. **Security:** Computer networks provide security through authorization. Authorization is done via user id and password. So, it ensures that when we log in, we are only able to do it when there is a perfect match between our details and the details stored in the database.

23. Polymorphism: Polymorphism is the ability to use an operator or function in more than one way. The methods in the child class that have the same name as the methods in the parent class behave differently.

    Operator Overloading: The feature where and operator can be used in different forms is known as operator overloading. Example the operator '+' behaves differently for integers and strings. For integers it adds the numbers and for strings it concatenates two strings.

Function Overloading: It is the ability to create multiple functions of the same name with different implementations. Function name remains same but their arguments and data type varies.

24.
```
book_dict={ }
n = int(input("Enter the number of books: "))
for i in range(n):
    book_id=input("Enter Book ID: ")
    price = float(input("Enter Price: "))
    book_dict[book_id]=price

print(book_dict)
print("\nBooks with price above 250: ")
for id,price in book_dict.items():
    if price>250:
    print("Book ID: ",id)
```

**OR**

24. Two dictionaries can be merged into one by using update() method. It merges the key and value of one dictionary into the other and overwrites values of the same key.
Syntax: D1.update(D2)  # D1, D2 are dictionaries

Example:
```
D1={1:10,2:20,3:30}
D2={4:40,5:50}
D1.update(D2)
print(D1)
```
Output:
{1:10,2:20,3:30, 4:40,5:50}

**Section D**

25.
```
    student=dict()
    n=int(input("Enter total number of students"))
    i = 1
    while i<= n:
       name=input('Enter the Name: ')
```

```
    rno=input('Enter the Roll No.: ')
    student[name]=rno
    i = i + 1
studname= input('Enter the Name to search: ')

if studname in student:

    print(student[studname])

else:
    print("Student not found!")
```

26.
```
T1 = tuple()

n=int(input('Total number of values in the first tuple'))
for i in range(n):
  a=int(input('Enter the Elements: '))

  T1=T1+(a,)


T2=tuple()
m=int(input('Total number of values in the second tuple'))
for i in range(m):
  a=int(input('Enter the Elements: '))
  T2=T2+(a,)
print('First tuple: ',T1)
print('Second tuple: ',T2)
T1,T2 = T2,T1
print('After swapping:')
print('First tuple: ',T1)
print('Second tuple: ',T2)
```

**OR**

26.
```
t=tuple()
n=int(input('Total number of values in tuple: '))
for i in range(n):
   a=int(input('Enter elements: '))
   t = t+(a,)

b=int(input('Enter first element to be added:'))
t=t+(b,)
print('Maximum value of the tuple: ',max(t))
print('Minimum value of the tuple: ',min(t))
```

********