# Noida Institute of Engineering and Technology, Greater Noida

## Problem Solving using Python (ACSE0101)

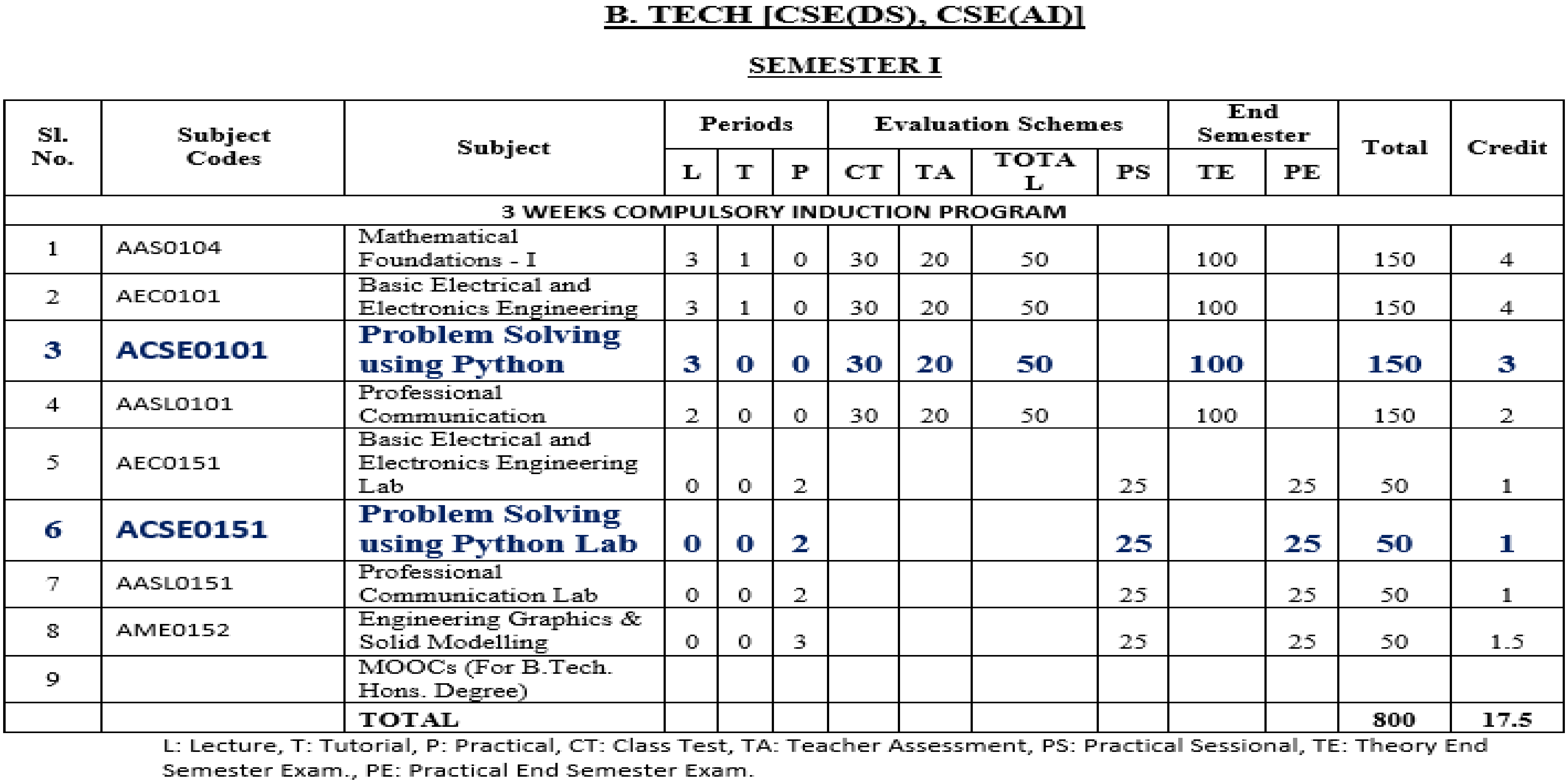
Unit: I

Basics of Python Programming

Course Details
(B. Tech. 1st Sem)

Utsav Kumar Malviya
(Asst. Professor)
CSE(IoT) Department

# Evaluation Scheme



## B. TECH [CSE(DS), CSE(AI)]

### SEMESTER I

| Sl. No. | Subject Codes | Subject | Periods | | | Evaluation Schemes | | | | End Semester | | Total | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | CT | TA | TOTA L | PS | TE | PE | | |
| 3 WEEKS COMPULSORY INDUCTION PROGRAM | | | | | | | | | | | | | |
| 1 | AAS0104 | Mathematical Foundations - I | 3 | 1 | 0 | 30 | 20 | 50 | | 100 | | 150 | 4 |
| 2 | AEC0101 | Basic Electrical and Electronics Engineering | 3 | 1 | 0 | 30 | 20 | 50 | | 100 | | 150 | 4 |
| **3** | **ACSE0101** | **Problem Solving using Python** | **3** | **0** | **0** | **30** | **20** | **50** | | **100** | | **150** | **3** |
| 4 | AASL0101 | Professional Communication | 2 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 2 |
| 5 | AEC0151 | Basic Electrical and Electronics Engineering Lab | 0 | 0 | 2 | | | | 25 | | 25 | 50 | 1 |
| **6** | **ACSE0151** | **Problem Solving using Python Lab** | **0** | **0** | **2** | | | | **25** | | **25** | **50** | **1** |
| 7 | AASL0151 | Professional Communication Lab | 0 | 0 | 2 | | | | 25 | | 25 | 50 | 1 |
| 8 | AME0152 | Engineering Graphics & Solid Modelling | 0 | 0 | 3 | | | | 25 | | 25 | 50 | 1.5 |
| 9 | | MOOCs (For B.Tech. Hons. Degree) | | | | | | | | | | | |
| | | **TOTAL** | | | | | | | | | | **800** | **17.5** |

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., PE: Practical End Semester Exam.

# Syllabus

## UNIT-I: Basics of python programming

**Introduction:** Introduction to computer system, algorithms and flowcharts, Ethics and IT policy in company, A Brief History of Python, Applications areas of python, The Programming Cycle for Python, Python IDE.

**Elements of Python:** Keywords and identifiers, variables, data types and type conversion, operators in python, Operator precedence and associativity, expressions in python, strings, Indexing and Slicing of Strings, Classes and object, constructor.

# Syllabus

## UNIT-II: Decision Control Statements

**Conditional Statements**: if statement, if-else statement, Nested-if statement and elif statements.

**Loops:** Purpose and working of loops, while loop, for loop, else with loop statement, Selecting an appropriate loop, Nested Loops, break, continue and pass statement.

# Syllabus

## UNIT-III: Function and Modules

Introduction of Function, calling a function, Function arguments, Mutability and Immutability, built in function, scope rules, Namespaces, Garbage Collection, Passing function to a function, recursion, Lambda functions, Map, filter, Reduce.

**Modules and Packages:** Importing Modules, writing own modules, Standard library modules, dir( ) Function, Packages in Python.

# Syllabus

## UNIT-IV: Basic Data structures in Python

**Python Basic Data Structures:** Sequence, Packing and Unpacking Sequences, Mutable Sequences, Strings, Basic operations, Comparing strings, string formatting, Slicing, Built-in string methods and function, Regular expressions, Lists, Tuples, Sets and Dictionaries with built-in methods, List Comprehension, Looping in basic data structures.

# Syllabus

## UNIT-V: File and Exception handling

**Files and Directories:** Introduction to File Handling, Reading and Writing files, Additional file methods, Working with Directories.

**Exception Handling**, Errors, Run Time Errors, Handling I/O Exception, Try-except statement, Raise, Assert.

# Course Objective

In this semester, the students will

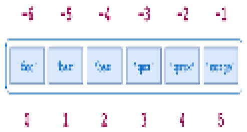
- Study the basic of computers and algorithms.
- Study the history of Python and Programming Cycle for Python.
- Gain the understanding of the Python IDE and understand how to write Python Program.
- Learn the elements of Python like Keywords and Identifiers, Variables, Data types and Operators.

# Course Objective

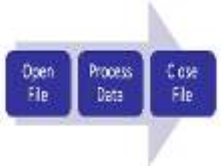
Learn the conditional statement like simple if, if-else, nested if-else and elif ladder.

Study the loop statement like while and for loop.

Gain the understanding of functions, modules and packages in python.



Learn the basic python data structures like string, list, tuple, set and dictionary.



Study the file and exception handling to develop real life application.

# Course Outcome (CO)

| Course Outcome ( CO) | At the end of course , the student will be able to: | Bloom's Knowledge Level (KL) |
|---|---|---|
| CO1 | Analyse and implement simple python programs. | K3, K4 |
| CO2 | Develop Python programs using decision control statements. | K3, K6 |
| CO3 | Implement user defined functions and modules in python. | K2 |
| CO4 | Implement programs using python data structures –string, lists, tuples, set, dictionaries. | K3 |
| CO5 | Perform input/output operations with files in python and apply exception handling for uninterrupted execution. | K3, K4 |

# Program Outcomes (POs)

**Engineering Graduates will be able to:**

**PO1 : Engineering Knowledge**

**PO2 : Problem Analysis**

**PO3 : Design/Development of solutions**

**PO4 : Conduct Investigations of complex problems**

**PO5 : Modern tool usage**

**PO6 : The engineer and society**

# Program Outcomes (POs)

**Engineering Graduates will be able to:**

**PO7 : Environment and sustainability**

**PO8 : Ethics**

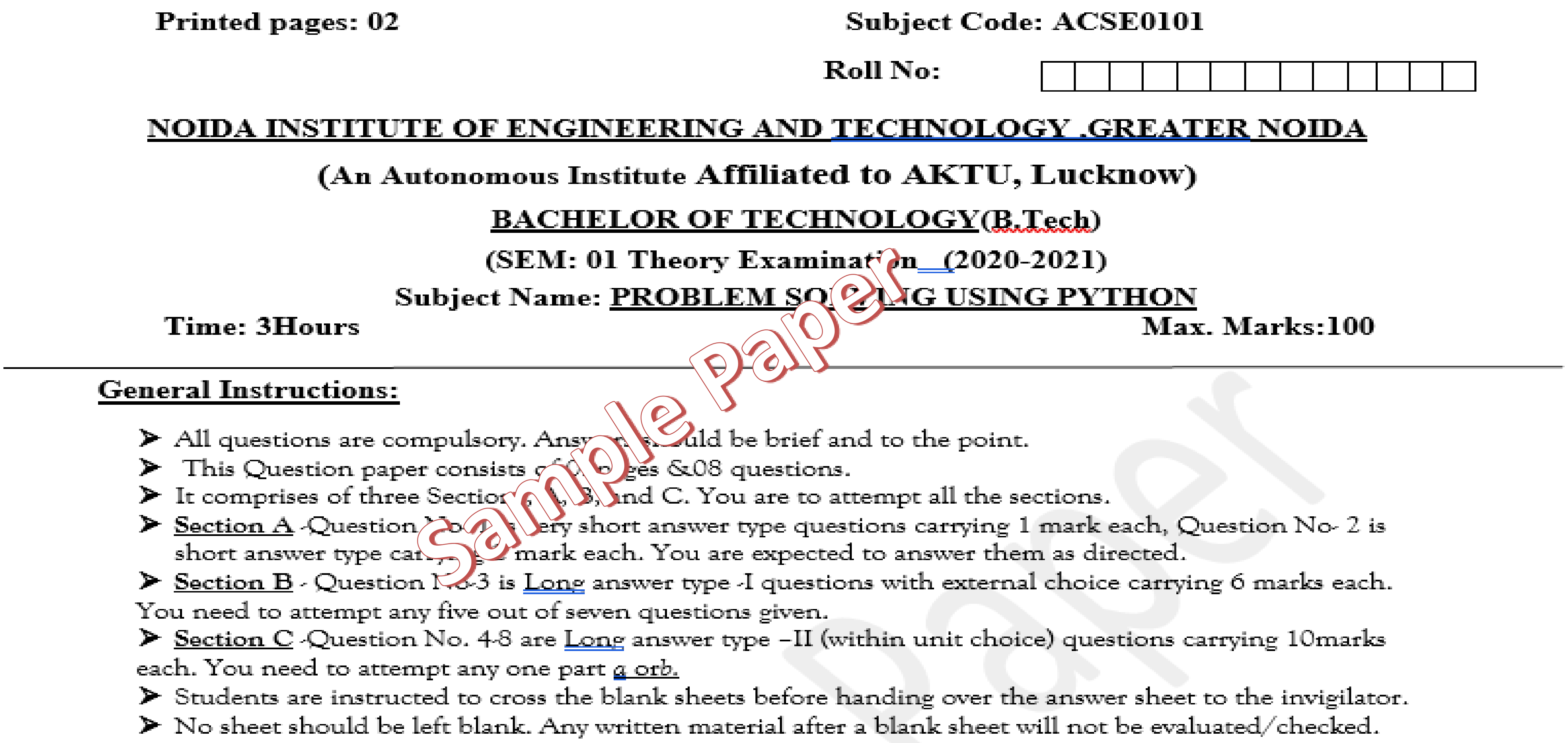
**PO9 : Individual and teamwork**

**PO10 : Communication**

**PO11 : Project management and finance**

**PO12 : Life-long learning**

# Pattern of Offline External Exam Question Paper (100 marks)



**Printed pages: 02** **Subject Code: ACSE0101**

**Roll No:** □□□□□□□□□□□□□

**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY ,GREATER NOIDA**

**(An Autonomous Institute Affiliated to AKTU, Lucknow)**

**BACHELOR OF TECHNOLOGY(B.Tech)**

**(SEM: 01 Theory Examination (2020-2021)**

**Subject Name: PROBLEM SOLVING USING PYTHON**

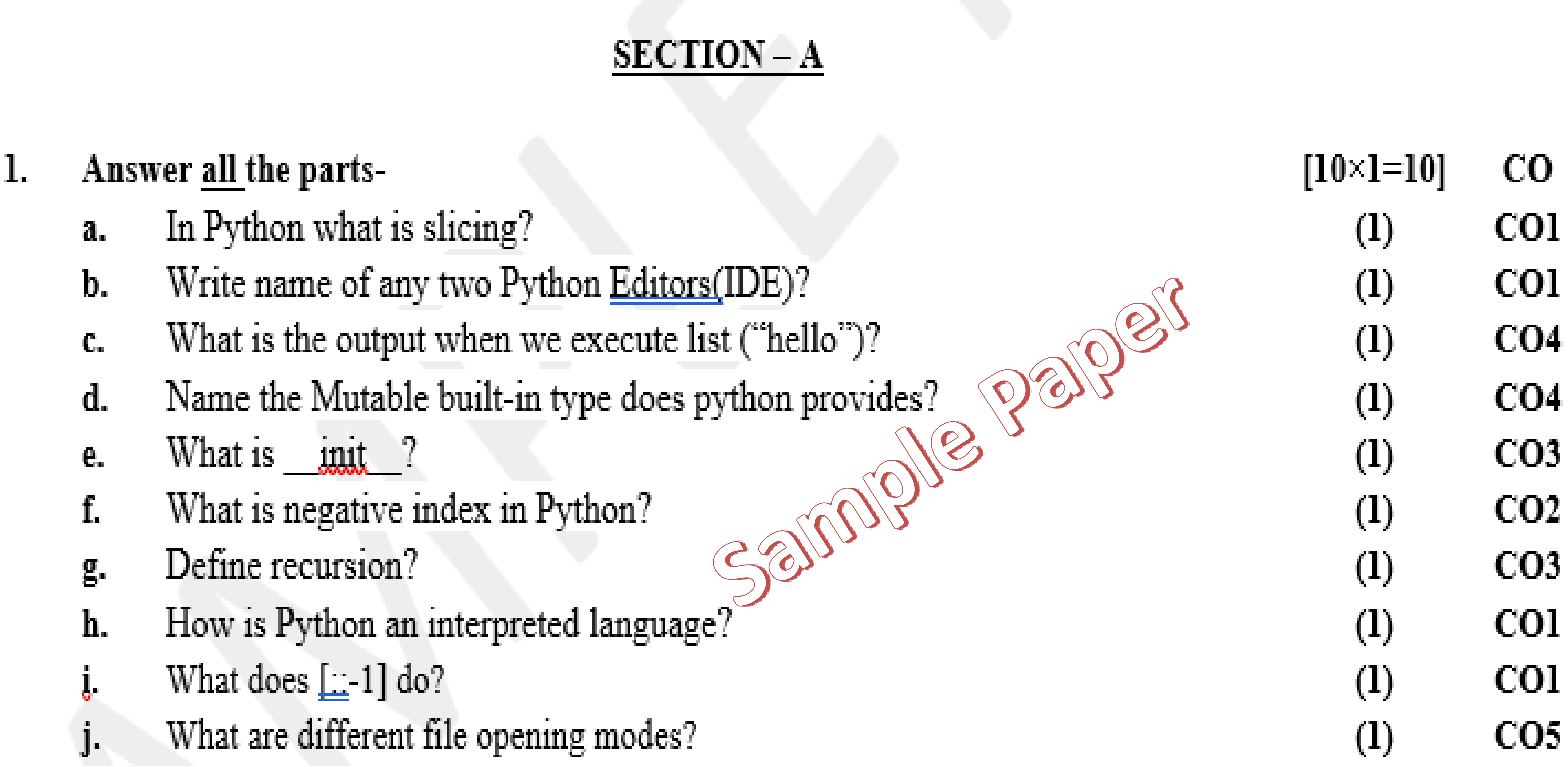
**Time: 3Hours** **Max. Marks:100**

**General Instructions:**

- All questions are compulsory. Answers should be brief and to the point.
- This Question paper consists of 02 pages &08 questions.
- It comprises of three Sections, A, B, and C. You are to attempt all the sections.
- Section A -Question No- 1 is very short answer type questions carrying 1 mark each, Question No- 2 is short answer type carrying 2 mark each. You are expected to answer them as directed.
- Section B - Question No-3 is Long answer type -I questions with external choice carrying 6 marks each. You need to attempt any five out of seven questions given.
- Section C -Question No. 4-8 are Long answer type –II (within unit choice) questions carrying 10marks each. You need to attempt any one part a orb.
- Students are instructed to cross the blank sheets before handing over the answer sheet to the invigilator.
- No sheet should be left blank. Any written material after a blank sheet will not be evaluated/checked.

Sample Paper

# Pattern of Offline External Exam Question Paper (100 marks)

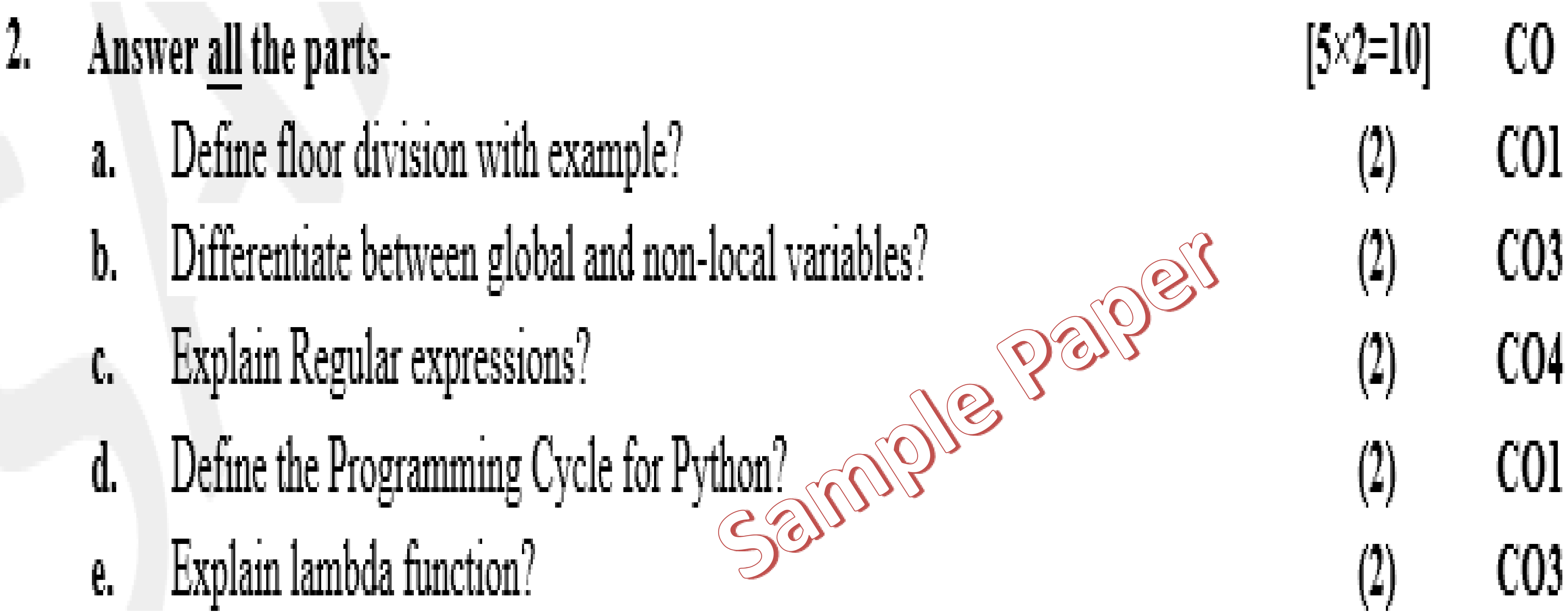


## SECTION – A

| | | | |
|---|---|---|---|
| 1. | Answer **all** the parts- | [10×1=10] | CO |
| a. | In Python what is slicing? | (1) | CO1 |
| b. | Write name of any two Python Editors(IDE)? | (1) | CO1 |
| c. | What is the output when we execute list (“hello”)? | (1) | CO4 |
| d. | Name the Mutable built-in type does python provides? | (1) | CO4 |
| e. | What is __init__? | (1) | CO3 |
| f. | What is negative index in Python? | (1) | CO2 |
| g. | Define recursion? | (1) | CO3 |
| h. | How is Python an interpreted language? | (1) | CO1 |
| i. | What does [::-1] do? | (1) | CO1 |
| j. | What are different file opening modes? | (1) | CO5 |

Sample Paper

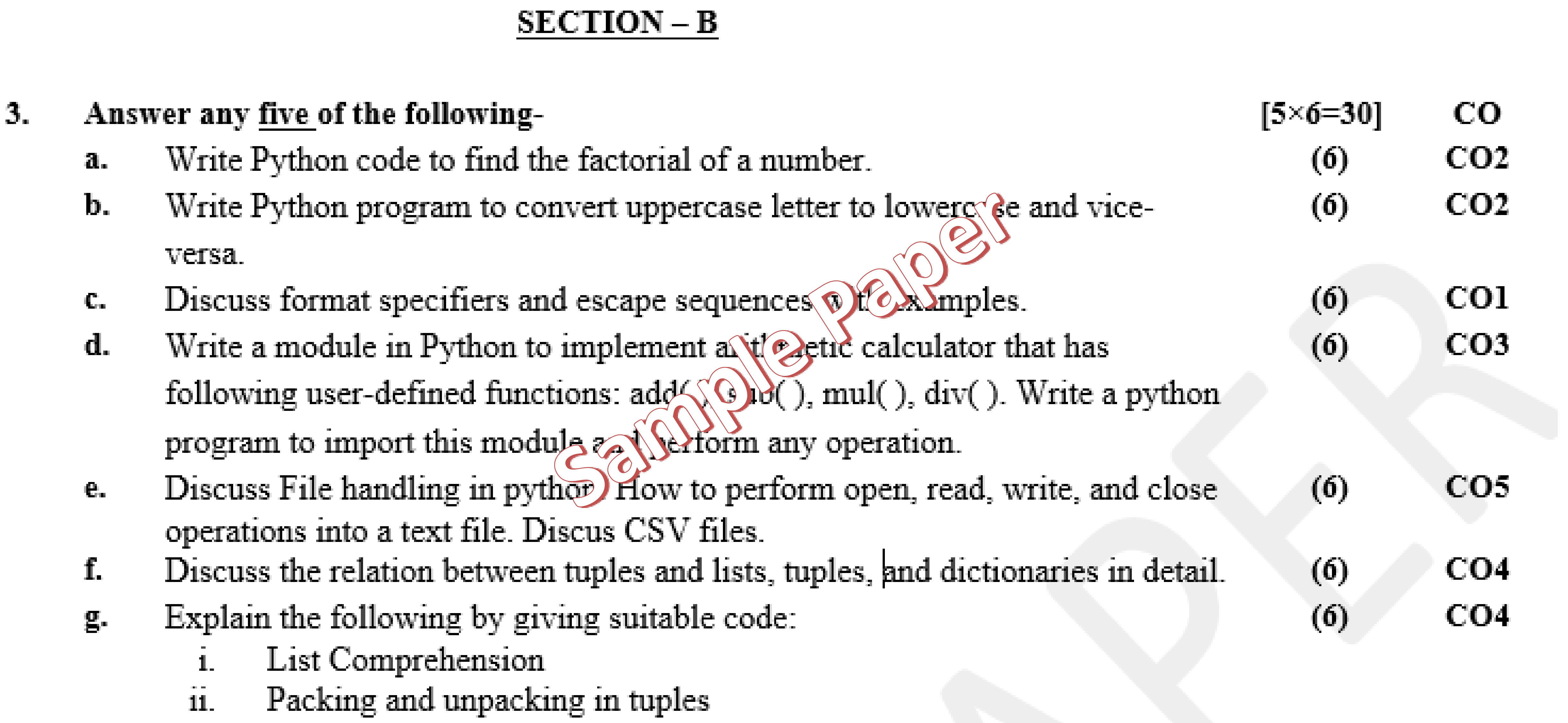
# Pattern of Offline External Exam Question Paper (100 marks)



| | | | |
|---|---|---|---|
| 2. | Answer all the parts- | [5×2=10] | CO |
| a. | Define floor division with example? | (2) | CO1 |
| b. | Differentiate between global and non-local variables? | (2) | CO3 |
| c. | Explain Regular expressions? | (2) | CO4 |
| d. | Define the Programming Cycle for Python? | (2) | CO1 |
| e. | Explain lambda function? | (2) | CO3 |

Sample Paper

# Pattern of Offline External Exam Question Paper (100 marks)

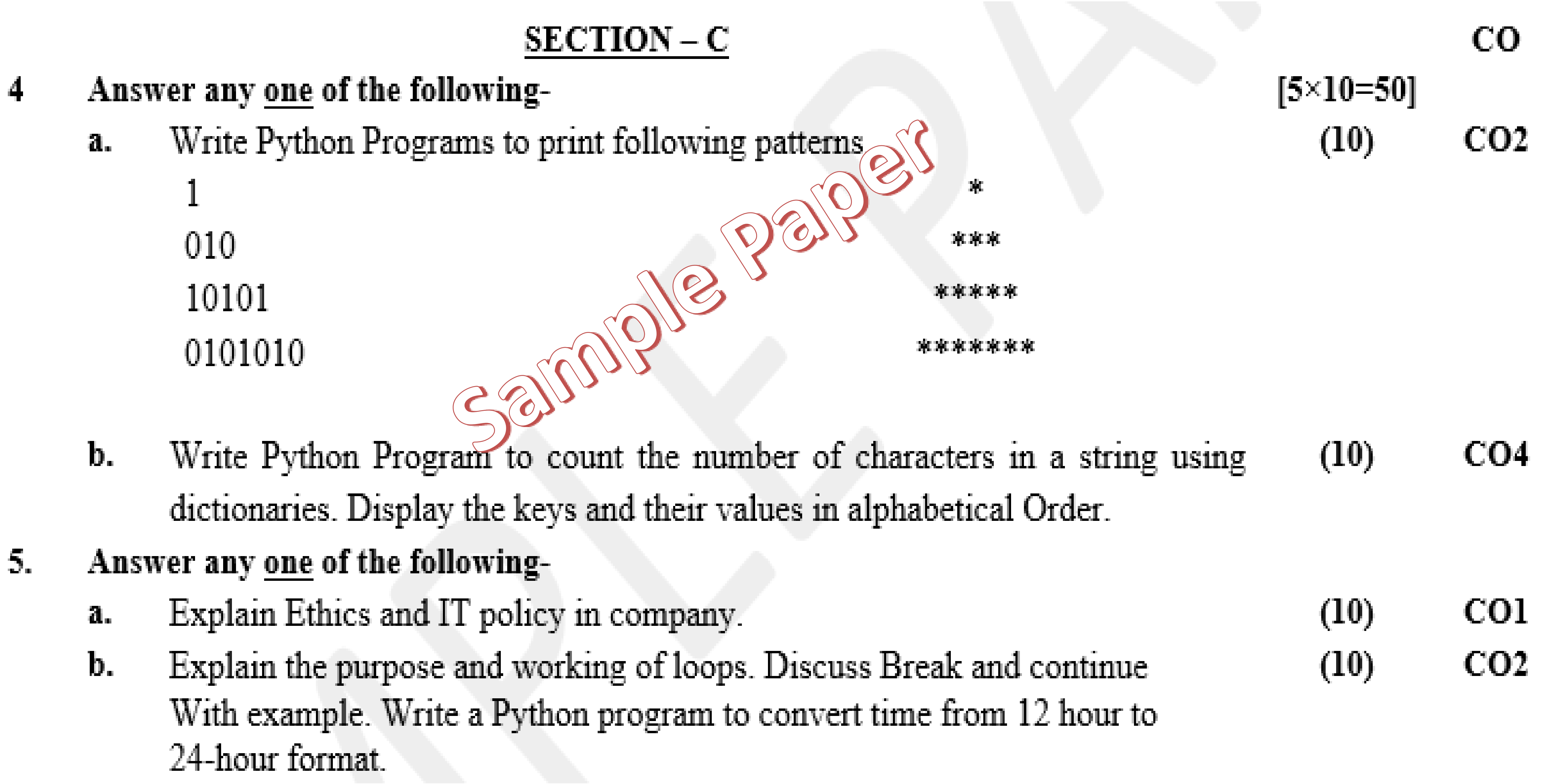


## SECTION – B

| | | | | |
|---|---|---|---|---|
| **3.** | **Answer any five of the following-** | | **[5×6=30]** | **CO** |
| | **a.** | Write Python code to find the factorial of a number. | **(6)** | **CO2** |
| | **b.** | Write Python program to convert uppercase letter to lowercase and vice-versa. | **(6)** | **CO2** |
| | **c.** | Discuss format specifiers and escape sequences with examples. | **(6)** | **CO1** |
| | **d.** | Write a module in Python to implement arithmetic calculator that has following user-defined functions: add( ), sub( ), mul( ), div( ). Write a python program to import this module and perform any operation. | **(6)** | **CO3** |
| | **e.** | Discuss File handling in python. How to perform open, read, write, and close operations into a text file. Discus CSV files. | **(6)** | **CO5** |
| | **f.** | Discuss the relation between tuples and lists, tuples, and dictionaries in detail. | **(6)** | **CO4** |
| | **g.** | Explain the following by giving suitable code:<br>i. List Comprehension<br>ii. Packing and unpacking in tuples | **(6)** | **CO4** |

Sample Paper
PAPER

# Pattern of Offline External Exam Question Paper (100 marks)



**SECTION – C**      **CO**

**4**   **Answer any one of the following-**   **[5×10=50]**

**a.** Write Python Programs to print following patterns   (10)   CO2

```
1                             *
010                          ***
10101                       *****
0101010                    *******
```

**b.** Write Python Program to count the number of characters in a string using dictionaries. Display the keys and their values in alphabetical Order.   (10)   CO4

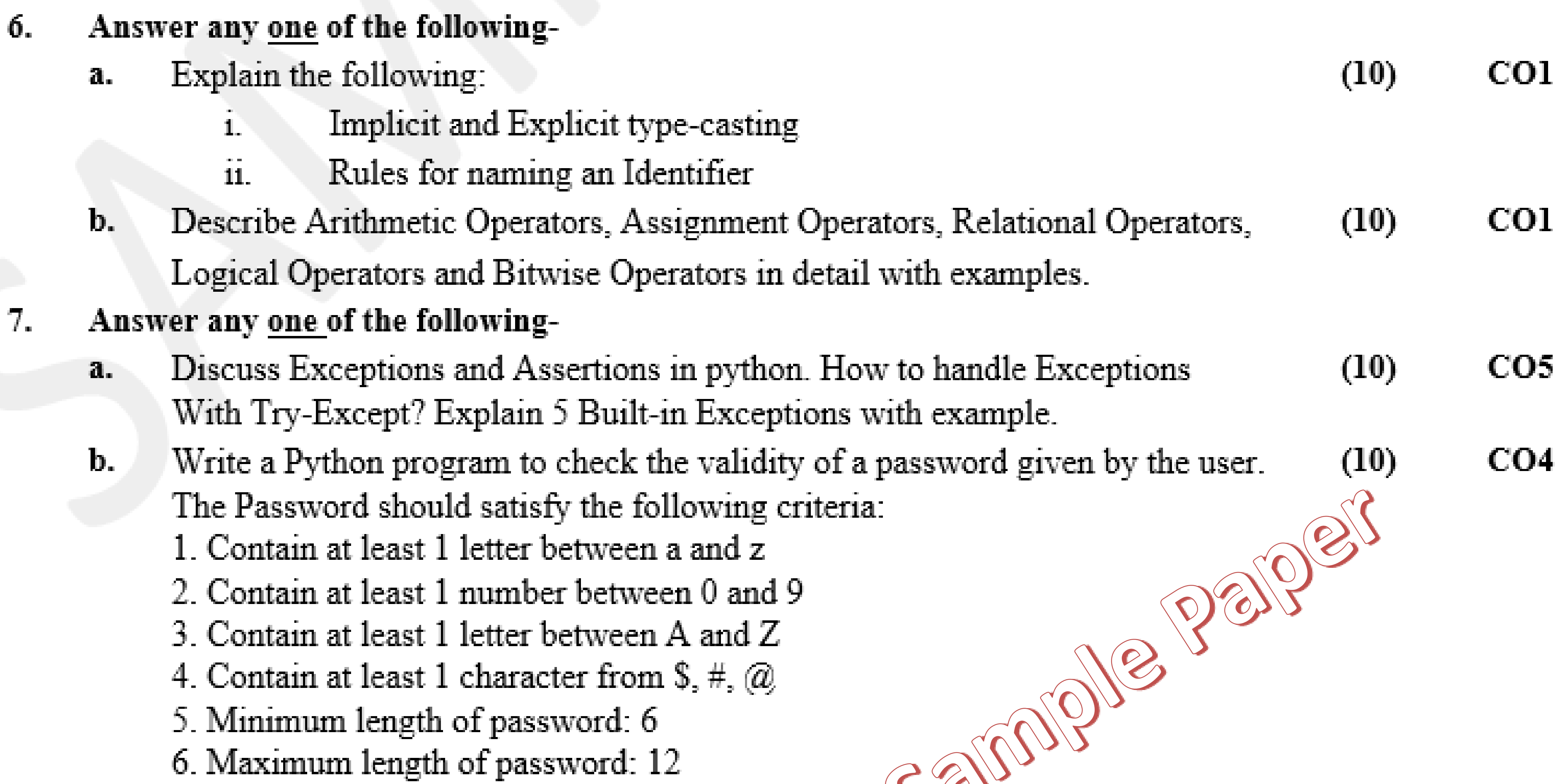
**5.**   **Answer any one of the following-**

**a.** Explain Ethics and IT policy in company.   (10)   CO1

**b.** Explain the purpose and working of loops. Discuss Break and continue With example. Write a Python program to convert time from 12 hour to 24-hour format.   (10)   CO2

Sample Paper

# Pattern of Offline External Exam Question Paper (100 marks)



6. **Answer any <u>one</u> of the following-**

a. Explain the following: (10) CO1

i. Implicit and Explicit type-casting

ii. Rules for naming an Identifier

b. Describe Arithmetic Operators, Assignment Operators, Relational Operators, Logical Operators and Bitwise Operators in detail with examples. (10) CO1

7. **Answer any <u>one</u> of the following-**

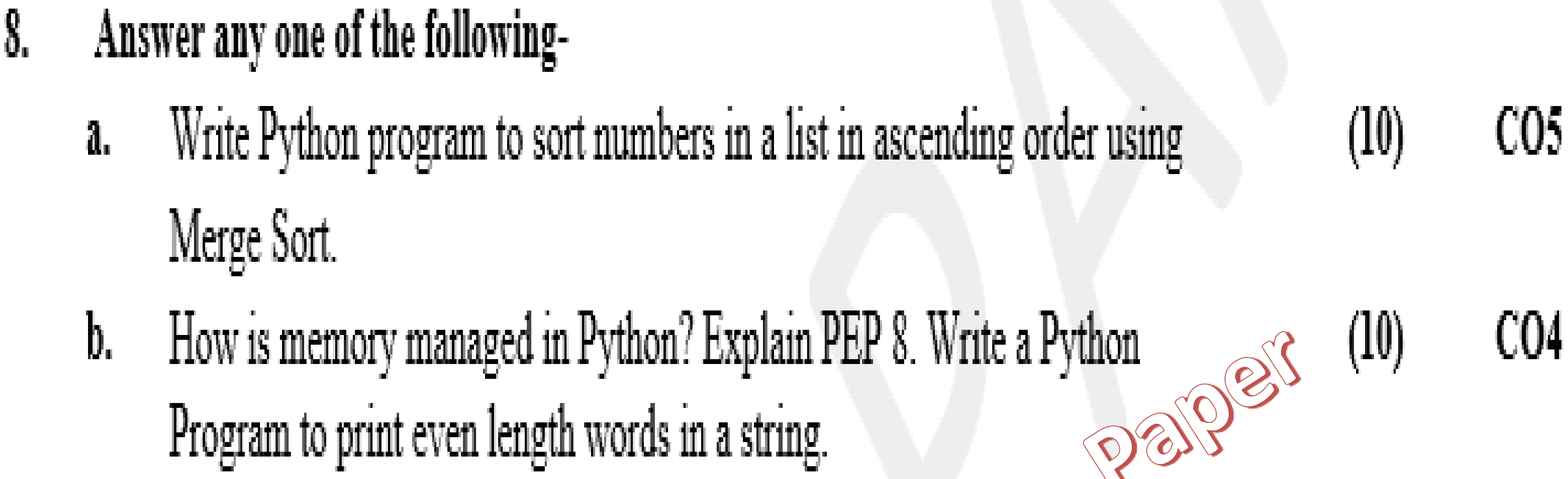
a. Discuss Exceptions and Assertions in python. How to handle Exceptions With Try-Except? Explain 5 Built-in Exceptions with example. (10) CO5

b. Write a Python program to check the validity of a password given by the user. The Password should satisfy the following criteria: (10) CO4

1. Contain at least 1 letter between a and z
2. Contain at least 1 number between 0 and 9
3. Contain at least 1 letter between A and Z
4. Contain at least 1 character from $, #, @
5. Minimum length of password: 6
6. Maximum length of password: 12

Sample Paper

# Pattern of Offline External Exam Question Paper (100 marks)



8. Answer any one of the following-

| | | | |
|---|---|---|---|
| a. | Write Python program to sort numbers in a list in ascending order using Merge Sort. | (10) | CO5 |
| b. | How is memory managed in Python? Explain PEP 8. Write a Python Program to print even length words in a string. | (10) | CO4 |

Sample Paper

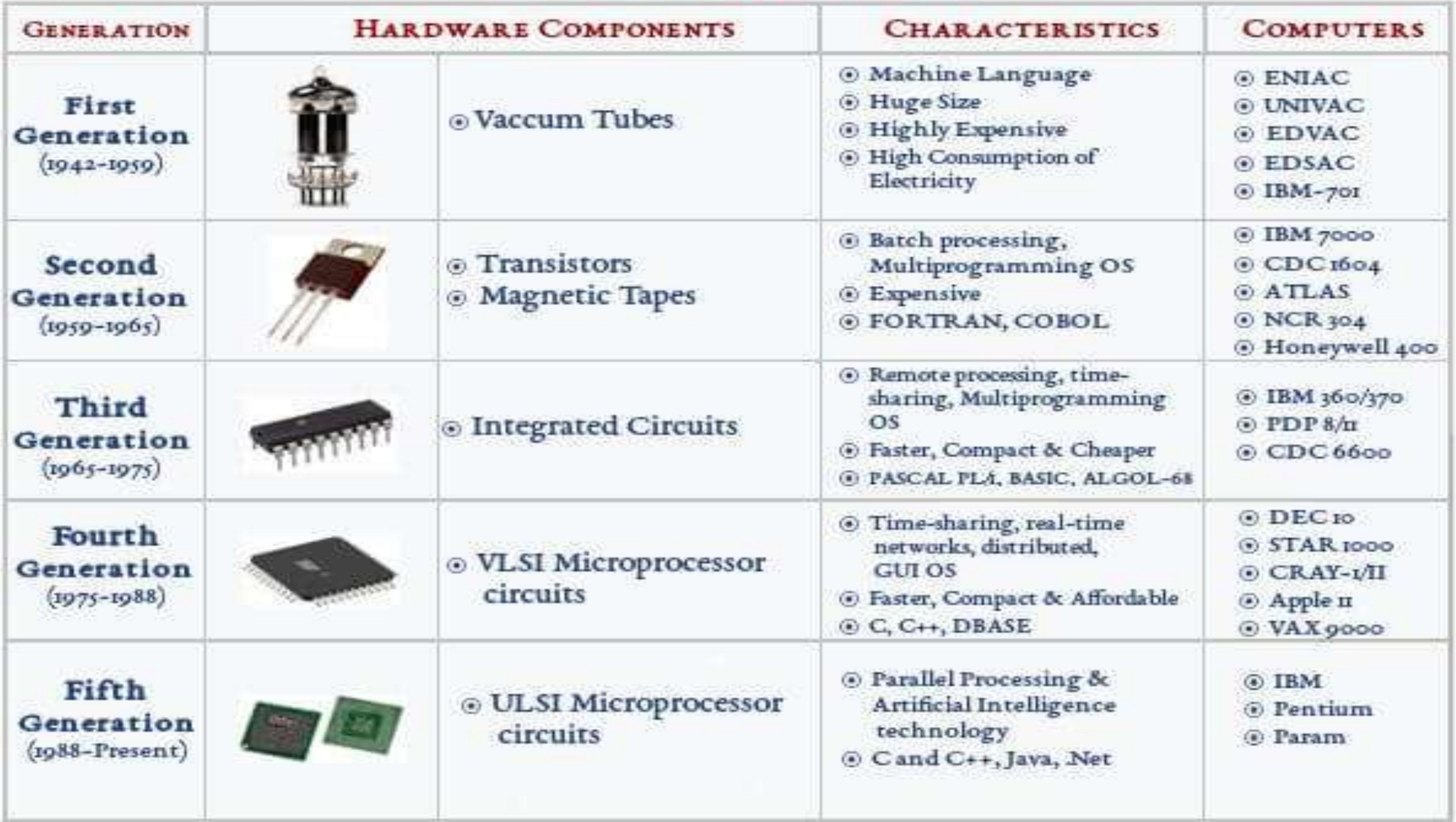
# What is Python Language?

- Python is a high-level general-purpose, interpreted, interactive, object-oriented and reliable language having wide range of applications from Web development, scientific and mathematical computing to desktop graphical user Interfaces.
- The syntax of the language is clean, and length of the code is relatively short.
- It allows to think about the problem rather than focusing on the syntax

# Difference between Python and C Language

| Python Language | C Language |
|---|---|
| 1. Python is an interpreted, high-level, general-purpose programming language. | 1. C is a general-purpose compiled, procedural computer programming language. |
| 2. Interpreted programs execute slower as compared to compiled programs. | 2. Compiled programs execute faster as compared to interpreted programs. |
| 3. It is easier to write a code in Python as the number of lines is less comparatively. | 3. Program syntax is harder than Python. |
| 4. No pointers functionality available in Python | 4. Pointers are available in C. |

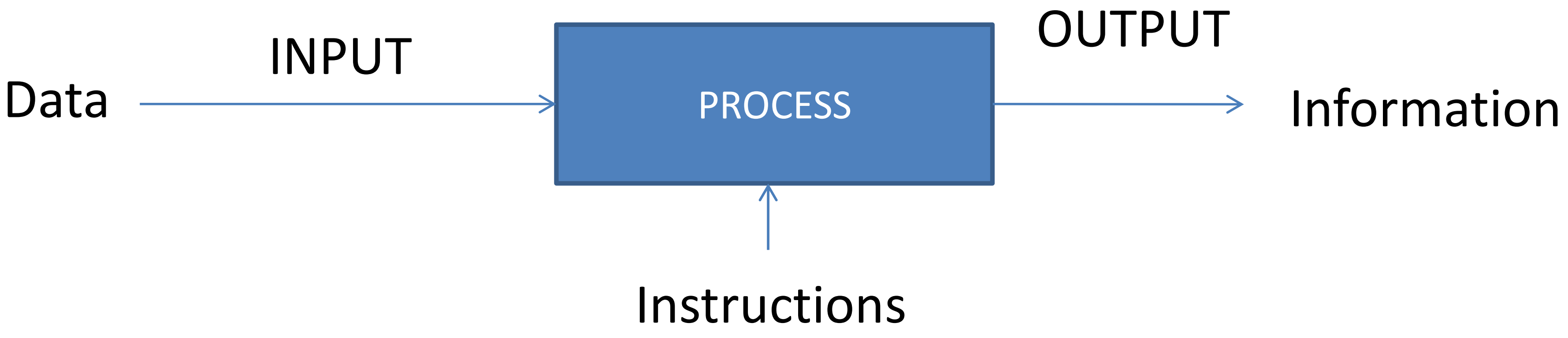
# COMPUTER GENERATIONS



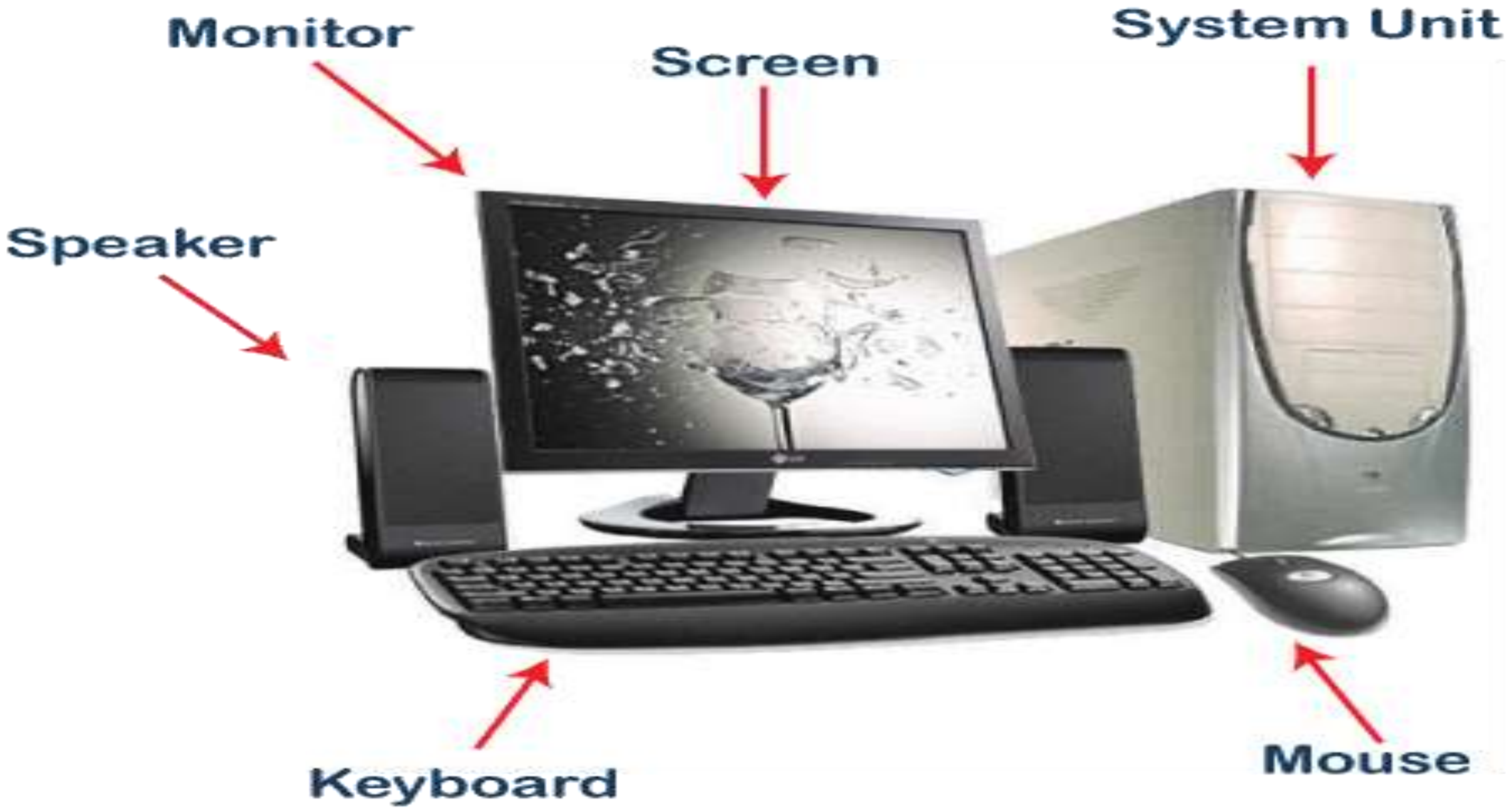
| GENERATION | HARDWARE COMPONENTS | | CHARACTERISTICS | COMPUTERS |
|---|---|---|---|---|
| **First Generation** (1942-1959) | | ⊙ Vaccum Tubes | ⊙ Machine Language<br>⊙ Huge Size<br>⊙ Highly Expensive<br>⊙ High Consumption of Electricity | ⊙ ENIAC<br>⊙ UNIVAC<br>⊙ EDVAC<br>⊙ EDSAC<br>⊙ IBM-701 |
| **Second Generation** (1959-1965) | | ⊙ Transistors<br>⊙ Magnetic Tapes | ⊙ Batch processing, Multiprogramming OS<br>⊙ Expensive<br>⊙ FORTRAN, COBOL | ⊙ IBM 7000<br>⊙ CDC 1604<br>⊙ ATLAS<br>⊙ NCR 304<br>⊙ Honeywell 400 |
| **Third Generation** (1965-1975) | | ⊙ Integrated Circuits | ⊙ Remote processing, time-sharing, Multiprogramming OS<br>⊙ Faster, Compact & Cheaper<br>⊙ PASCAL PL/I, BASIC, ALGOL-68 | ⊙ IBM 360/370<br>⊙ PDP 8/11<br>⊙ CDC 6600 |
| **Fourth Generation** (1975-1988) | | ⊙ VLSI Microprocessor circuits | ⊙ Time-sharing, real-time networks, distributed, GUI OS<br>⊙ Faster, Compact & Affordable<br>⊙ C, C++, DBASE | ⊙ DEC 10<br>⊙ STAR 1000<br>⊙ CRAY-1/II<br>⊙ Apple 11<br>⊙ VAX 9000 |
| **Fifth Generation** (1988-Present) | | ⊙ ULSI Microprocessor circuits | ⊙ Parallel Processing & Artificial Intelligence technology<br>⊙ C and C++, Java, .Net | ⊙ IBM<br>⊙ Pentium<br>⊙ Param |

# Computer (CO1)

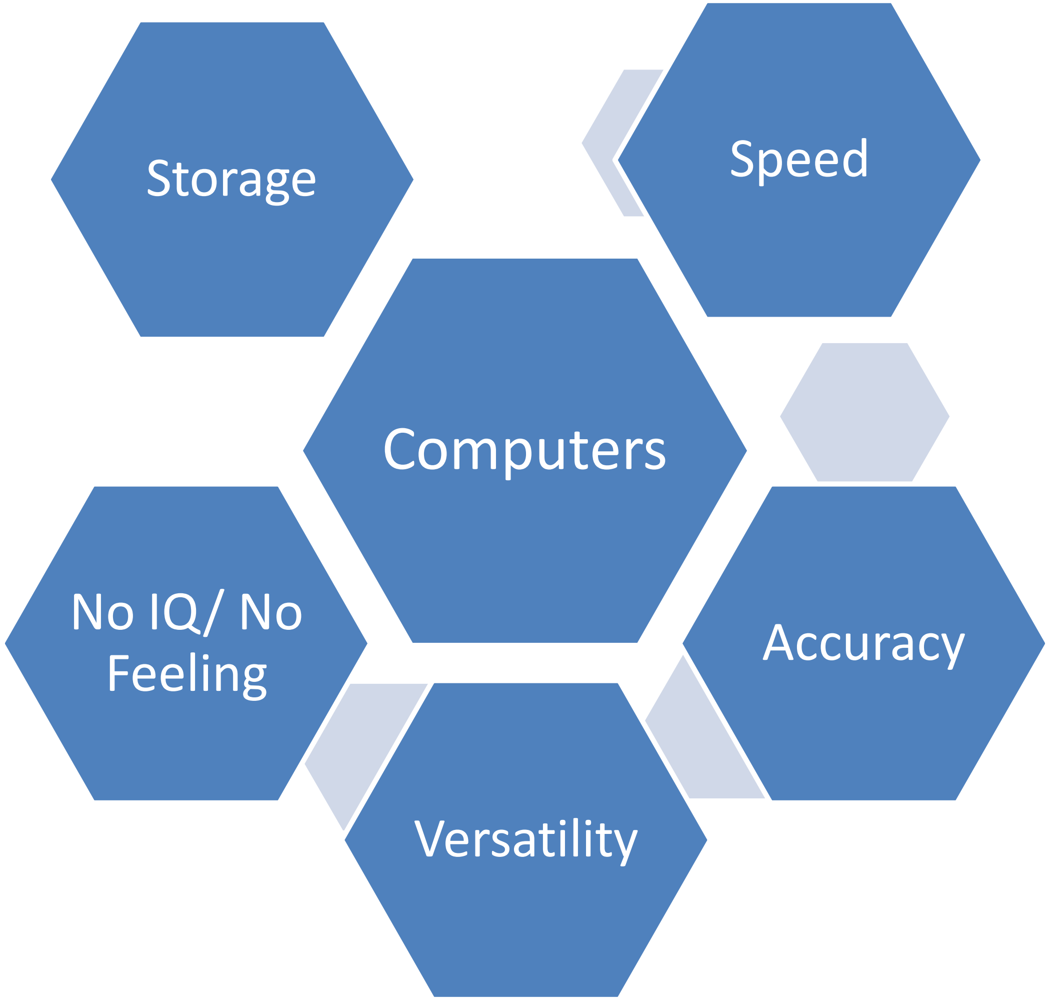
The term computer is derived from the term "**compute**". Computer is a **programmable electronic device** that takes data and instruction as an input from the user and, process data, and provides useful information.



# Computer (CO1)



# Characteristics of Computers (CO1)



Computers

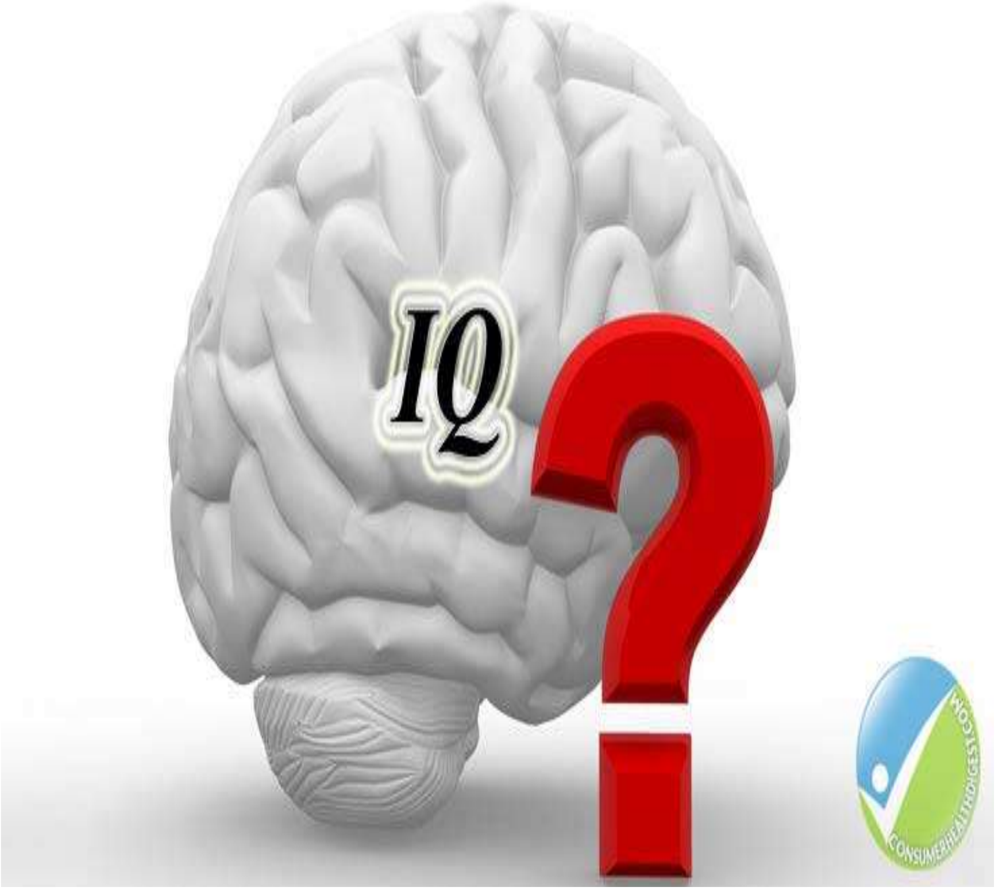
Storage

Speed

Accuracy

Versatility

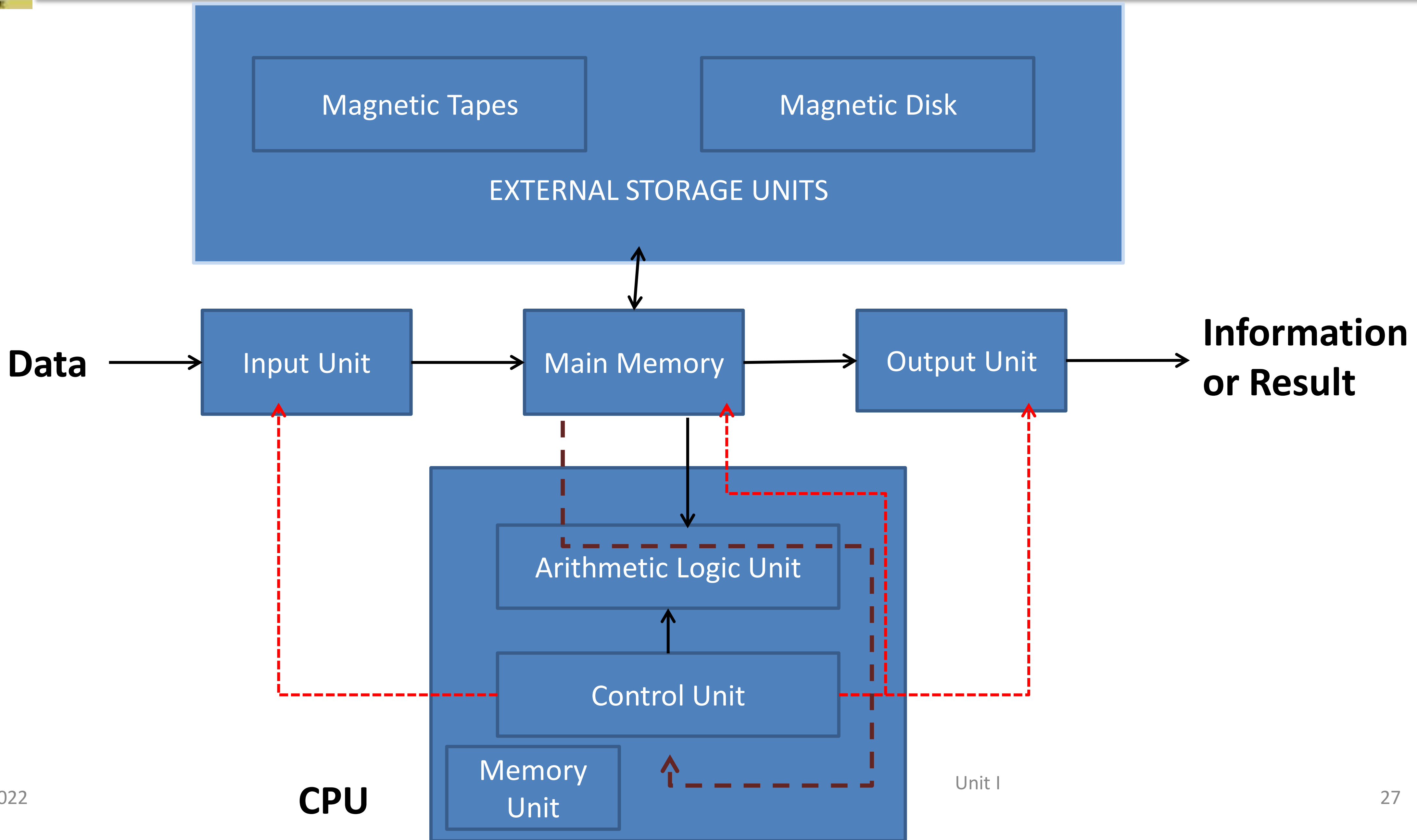
No IQ/ No Feeling



# Application of Computers (CO1)

- **Word Processing**
- **Internet**
- **Digital Audio/Video Compression**
- **Desktop Publishing**
- **Traffic Control**
- **Retail Business**
- **Hospitals**
- **Business and Industry**
- **Weather Forecasting**
- **Education**
- **Online Banking**
- **Robotics**
- **Expert Systems**

# Block Diagram of Digital Computers (CO1)



# Algorithm (CO1)

Algorithm is the well-defined computational procedures that takes some values or set of values as input, process it and produces some values or set of values as output.

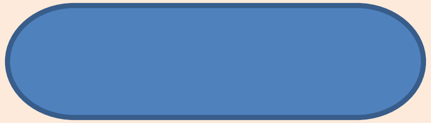
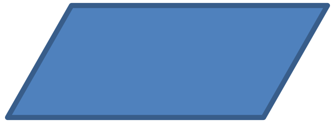
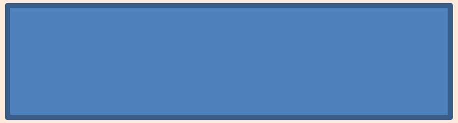
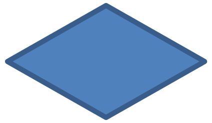
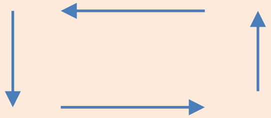
# Characteristics of Algorithm (CO1)

1. Input
   - It takes zero or more values as input.
2. Output
   - It produces one or more values as output.
3. Finiteness
   - It should terminate after finite number of steps.
4. Definiteness
   - Each instruction must be clear, precise and unambiguous.
5. Effectiveness
   - Each instruction must be very basic but essential so that it can be carried out using pen and paper.

# Flowchart (CO1)

- It is the pictorial representation of the algorithm.
- It is used by programmer as a programming planning tool for organizing the sequence of steps necessary to solve the problem.
- It is also known as the roadmap for programming.

# Elements of Flowchart (CO1)

| Picture | Name | Meaning |
|---|---|---|
|  | Ellipse/Oval | Start/End |
|  | Rhombus | Input and output |
|  | Rectangle | Data Processing |
|  | Diamond | Condition |
|  | Arrows | Movement or flow of operations |

# Algorithm and flowchart (CO1)

1. Write the algorithm and draw the flowchart to compute the sum of two numbers.
2. Write the algorithm and draw the flowchart to compute the sum and average of five numbers.
3. Write the algorithm and draw the flowchart to compute the area of triangle using heron’s formula.
4. Write the algorithm and draw the flowchart to compute the temperature in degree Fahrenheit when temperature in degree Celsius is given.
5. Write the algorithm and draw the flowchart to swap two numbers using 3rd variable.
6. Write the algorithm and draw the flowchart to swap two numbers without using 3rd variable.

# Algorithm and flowchart(CO1)

7. Write the algorithm and draw the flowchart to check whether the number is odd or even.
8. Write the algorithm and draw the flowchart to compute the greater of two numbers.
9. Write the algorithm and draw the flowchart to compute the greatest of three numbers.
10. Write the algorithm and draw the flowchart to check whether the given year is leap year or not.
11. Write the algorithm and draw the flowchart to compute the sum of first N natural numbers.

# Algorithm and flowchart(CO1)

12. Write the algorithm and draw the flowchart to compute the factorial of the given number.
13. Write the algorithm and draw the flowchart to compute the sum of digits of the given number.
14. Write the algorithm and draw the flowchart to compute the reverse of the given number. Also check the given number is in palindrome or not.
15. Write the algorithm and draw the flowchart to convert the decimal number to binary number.
16. Write the algorithm and draw the flowchart to convert the binary number to decimal number.

# Problem Solving using Python (ACSE0101)

Unit: I

## Basics of Python Programming

Course Details
(B. Tech. 1st Sem)

Utsav Kumar Malviya
(Asst. Professor)
CSE(IoT) Department

# Code of Ethics and Professional Conduct (CO1)

The Code is designed to inspire and guide the ethical conduct of all computing professionals, including current and aspiring practitioners, instructors, students, influencers, and anyone who uses computing technology in an impactful way.

# Code of Ethics and Professional Conduct (CO1)

**1. GENERAL ETHICAL PRINCIPLES**

*A computing professional should...*

- 1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.
- 1.2 Avoid harm.
- 1.3 Be honest and trustworthy.
- 1.4 Be fair and act not to discriminate.
- 1.5 Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.
- 1.6 Respect privacy.

# Code of Ethics and Professional Conduct… (CO1)

## 2. PROFESSIONAL RESPONSIBILITIES

*A computing professional should…*

- *2*.1 Strive to achieve high quality in both the processes and products of professional work.
- 2.2 Maintain high standards of professional competence, conduct, and ethical practice.
- 2.3 Know and respect existing rules pertaining to professional work.
- 2.4 Accept and provide appropriate professional review.

# Code of Ethics and Professional Conduct… (CO1)

## 2. PROFESSIONAL RESPONSIBILITIES

- 2.5 Foster public awareness and understanding of computing, related technologies, and their consequences.
- 2.6 Access computing and communication resources only when authorized or when compelled by the public good.
- 2.7 Design and implement systems that are robustly and usably secure.

# Code of Ethics and Professional Conduct… (CO1)

**3. PROFESSIONAL LEADERSHIP PRINCIPLES**

- 3.1 Ensure that the public good is the central concern during all professional computing work.
- 3.2 Manage personnel and resources to enhance the quality of working life.
- 3.3 Create opportunities for members of the organization or group to grow as professionals.

# Code of Ethics and Professional Conduct (CO1)

## 4.1 Uphold, promote, and respect the principles of the Code

## Programming Code of Ethics

Key points of proper conduct for Computer Programmers-
A programmer must...

- Never create or distribute malware.
- Never write code that is intentionally difficult to follow.
- Never write documentation that is intentionally confusing or inaccurate.
- Never reuse copyrighted code unless the proper license is purchased, or permission is obtained.

.

# Code of Ethics and Professional Conduct… (CO1)

- Acknowledge (verbally and in source code comments) the work of other programmers on which the code is based, even if substantial changes are made.
- Never intentionally introduce bugs with the intent of later claiming credit for fixing the bugs, or to stimulate the uptake of later versions.
- Never write code that intentionally breaks another programmer's code for the purpose of elevating one's status.
- Never hide known obstacles to a project's completion during any phase of development, especially the design phase.

# Code of Ethics and Professional Conduct… (CO1)

- Report any illegal activities of the employer.
- Never falsely deny the presence of bugs.
- Never reveal the secret corporate knowledge of an employer.
- Never accept compensation from multiple parties for the same work unless permission is given.
- Never conceal from the employer their financial interest in development resources.

# Code of Ethics and Professional Conduct… (CO1)

- Never maliciously injure the reputation of an employer or members of the development team.
- Never take credit for another's work.
- Never steal software, especially development tools.
- Never install third-party applications without the user's permission.

# IT Policy (CO1)

- Companies provides and maintains technological products, services and facilities like Personal Computers (PCs), peripheral equipment, servers, telephones, Internet and application software to its employees for official use.
- The Information Technology (IT) Policy of the organization defines rules, regulations and guidelines for proper usage and maintenance of these technological assets to ensure their ethical and acceptable use and assure health, safety and security of data, products, facilities as well as the people using them.
- It also provides guidelines for issues like purchase, compliance, IT support and grievance redressal of the employees pertaining to technological assets and services used for office work.

# IT Policy… (CO1)

- Acceptable Use Policy
- Security awareness
- DR/BCP (Disaster Recovery, Business Continuity plan)
- Change management.
- Equipment Usage policy
- PC standards
- The Internet Usage Policy
- Information security Policy
- Email and chat Policy
- The Software Usage Policy

# Python Introduction (CO1)

# Introduction (CO1)

## What is Python Language?

- Python is a high-level general-purpose, interpreted, interactive, object-oriented and reliable language having wide range of applications from Web development, scientific and mathematical computing to desktop graphical user Interfaces.
- The syntax of the language is clean, and length of the code is relatively short.
- It allows to think about the problem rather than focusing on the syntax

# A Brief History of Python (CO1)

- Guido Van Rossum was doing its application-based work in December of 1989 at Centrum Wiskunde & Informatica (CWI) which is situated in Netherland.
- It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas.

# A Brief History of Python Cont…(CO1)

- The programming language which Python is said to have succeeded is ABC Programming Language.
- He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features.
- After that what he did as very clever. He had taken the syntax of ABC, and some of its good features.

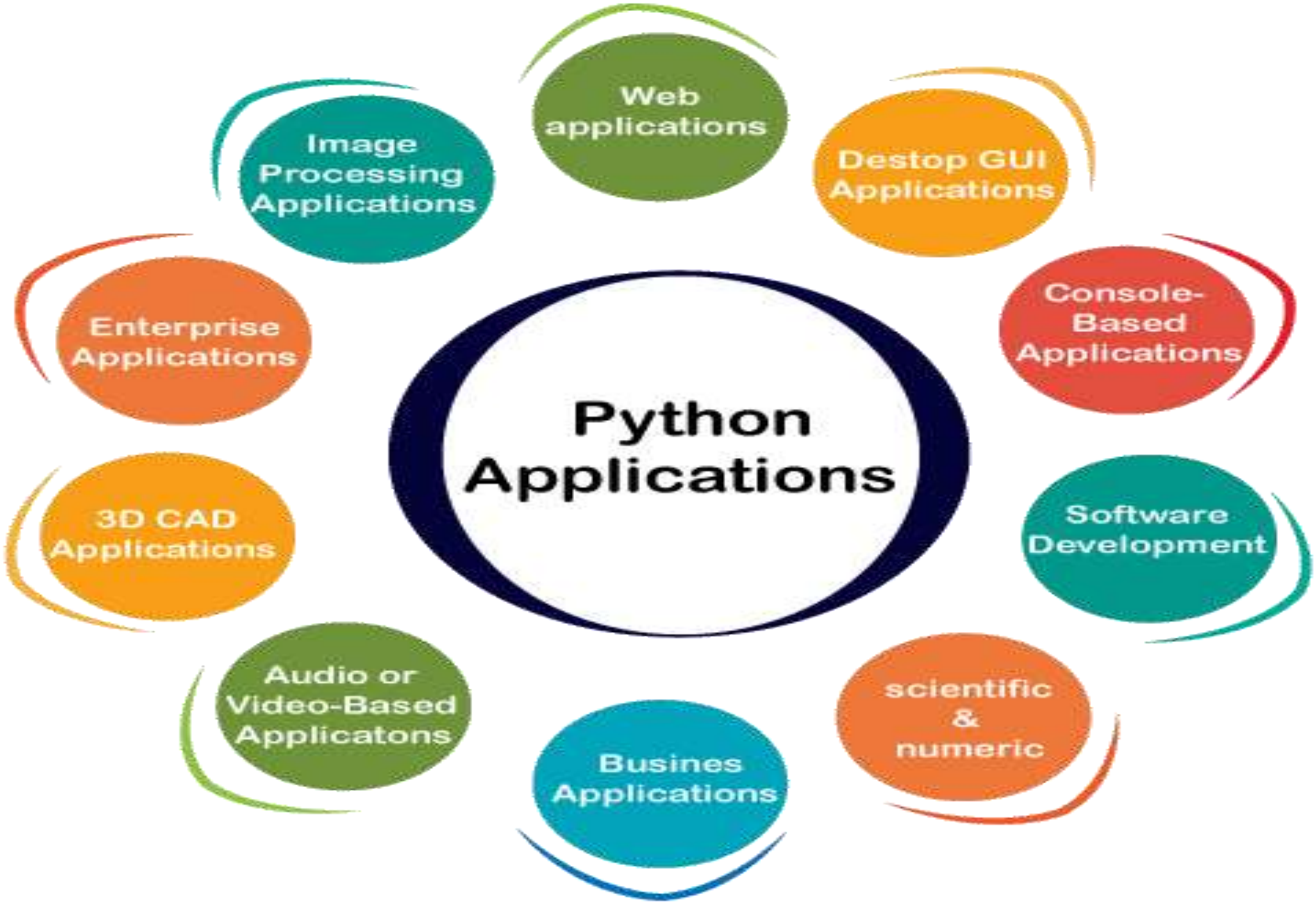
# A Brief History of Python Cont…(CO1)

- It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws.
- The inspiration for the name came from BBC’s TV Show – ‘Monty Python’s Flying Circus’, as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python!

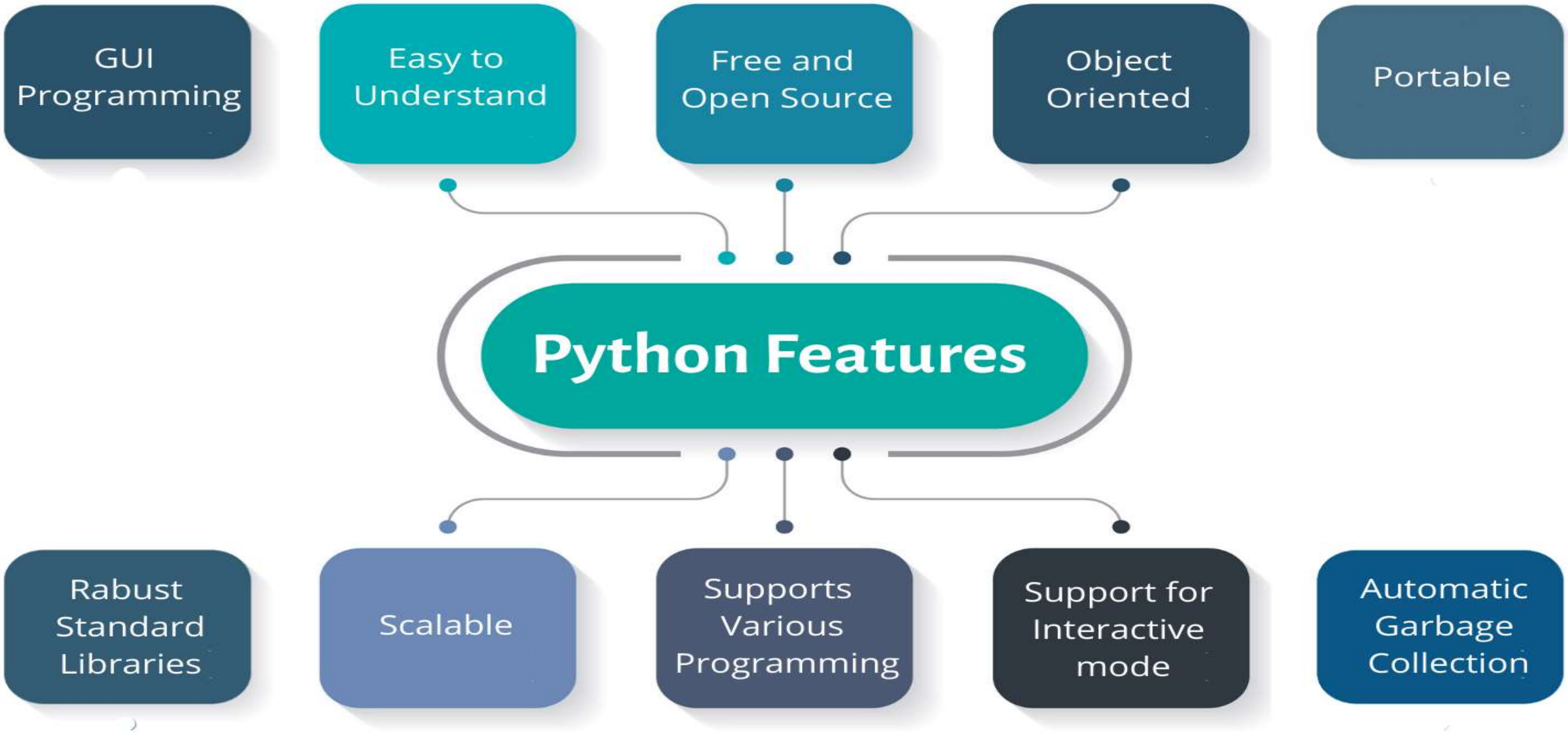
# A Brief History of Python Cont… (CO1)

- The language was finally released in 1991.
- When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C.
- Its main objective is to provide code readability and advanced developer productivity.

# Applications areas of python (CO1)



# Features of Python (CO1)



GUI Programming
Easy to Understand
Free and Open Source
Object Oriented
Portable
Python Features
Rabust Standard Libraries
Scalable
Supports Various Programming
Support for Interactive mode
Automatic Garbage Collection

# Features of Python (CO1)

- A simple language which is easier to learn.
- Free and open-source.
- Portability.
- Extensible and Embeddable
  - easily combine pieces of C/C++ or other languages with Python code.
- A high-level, interpreted language.
- Large standard libraries to solve common tasks.

# Features of Python (CO1)

- Object-oriented
  - Everything in Python is an object.
  - Object oriented programming (OOP) helps to solve a complex problem intuitively.
  - Structure supports such concepts as polymorphism, operation overloading, and multiple inheritance.
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

# Features of Python (CO1)

- It supports automatic garbage collection.
- **Scalable**
  - Python provides a better structure and support for large programs than shell scripting.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
- **Databases**
  - Python provides interfaces to all major commercial databases.

# Getting Python (CO1)

- The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python

  https://www.python.org/

- Python documentation can be downloaded from

  https://www.python.org/doc/