
# TERM-II


# Subject Teacher:- Mugdha K

# PROGRAMMING FOR PROBLEM SOVLING

# (PPS)

| Course Code | Name of Course | Course Category | Teaching Scheme | | | | Credits | Evaluation Scheme | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Theory | | | Practical | | Total Marks |
| | | | L | T | P | Total our | | TA | CAE | ESE | INT | EXT | |
| SEMESTER-II | | | | | | | | | | | | | |
| UBSL152 | Integral Calculus and Differential Equations | BS3 | 2 | 1 | - | 3 | 3 | 10 | 15 | 50 | - | - | 75 |
| UBSL153 | Linear Algebra and Statistics | BS4 | 2 | 1 | - | 3 | 3 | 10 | 15 | 50 | - | - | 75 |
| UBSL131/ UBSP131 | Environmental Chemistry | BS5 | 1 | | 2 | 3 | 2 | 10 | 15 | 50 | 25 | - | 100 |
| UCSP104 | Programming for Problem Solving | C4 | - | - | 4 | 4 | 2 | - | - | - | 50 | - | 50 |
| UECL104 / UECP104 | Modeling of Digital Circuits | C5 | 3 | - | 2 | 5 | 4 | 10 | 15 | 50 | 25 | - | 100 |
| UHUL101/ UHUP101 | Communication Skills | H1 | 2 | - | 2 | 4 | 3 | 10 | 15 | 50 | 25 | - | 100 |
| UECP105 | Internet of Things | A4 | - | - | 2 | 2 | 1 | - | - | - | 25 | - | 25 |
| | Foreign Language | A5 | - | - | 2 | 2 | 1 | - | - | - | 25 | - | 25 |
| TOTAL | | | 10 | 2 | 14 | 26 | 19 | | | | | | 550 |

| Unit | Contents | Hours |
|---|---|---|
| I | ALGORITHMIC PROBLEM SOLVING:<br>Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). | 8 |
| II | DATA, EXPRESSIONS, STATEMENTS<br>Python interpreter and interactive mode; values and types: int, float, Boolean, string, and list; variables, expressions, statements, Tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; | 8 |
| III | CONTROL FLOW, FUNCTIONS<br>Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, | 8 |

| | | |
|---|---|---|
| III | CONTROL FLOW, FUNCTIONS<br>Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, | 8 |
| IV | DICTIONARIES<br>Dictionaries: operations and methods; advanced list processing – list comprehension;<br>Object Oriented Programming: Classes and objects-inheritance-polymorphism | 8 |
| V | FILE HANDLING & EXCEPTION HANDLING<br>Overview of exception classes and Types: try, except, finally: File processing: reading and Writing files, Recent Trends in Python | 8 |

| Sr. No. | List of Experiment |
|---|---|
| 1 | Implementing if else in Python |
| 2 | Implementing loop in Python |
| 3 | Implementing Functions in Python |
| 4 | Implementing Set, Tuple & Dictionary in Python |
| 5 | Project Using Python<br>Module 1: Algorithms, Expression, Variables & I/O |
| 6 | Module 2: Control Structures |
| 7 | Module 3: List, Strings, Tuples & Dictionary |
| 8 | Module 4: Functions |
| 9 | Module 5: Object Oriented Programming |
| 10 | Module 6: Expression Handling & File Handling |

| | | |
|---|---|---|
| Text Books | 1. | Python Programming using problem solving Approach, By Reema Theraja, First Edition, 2017. |
| | 2. | A Byte of Python By C. H. Swaroop, Edition 2.1 |

# Free Courses

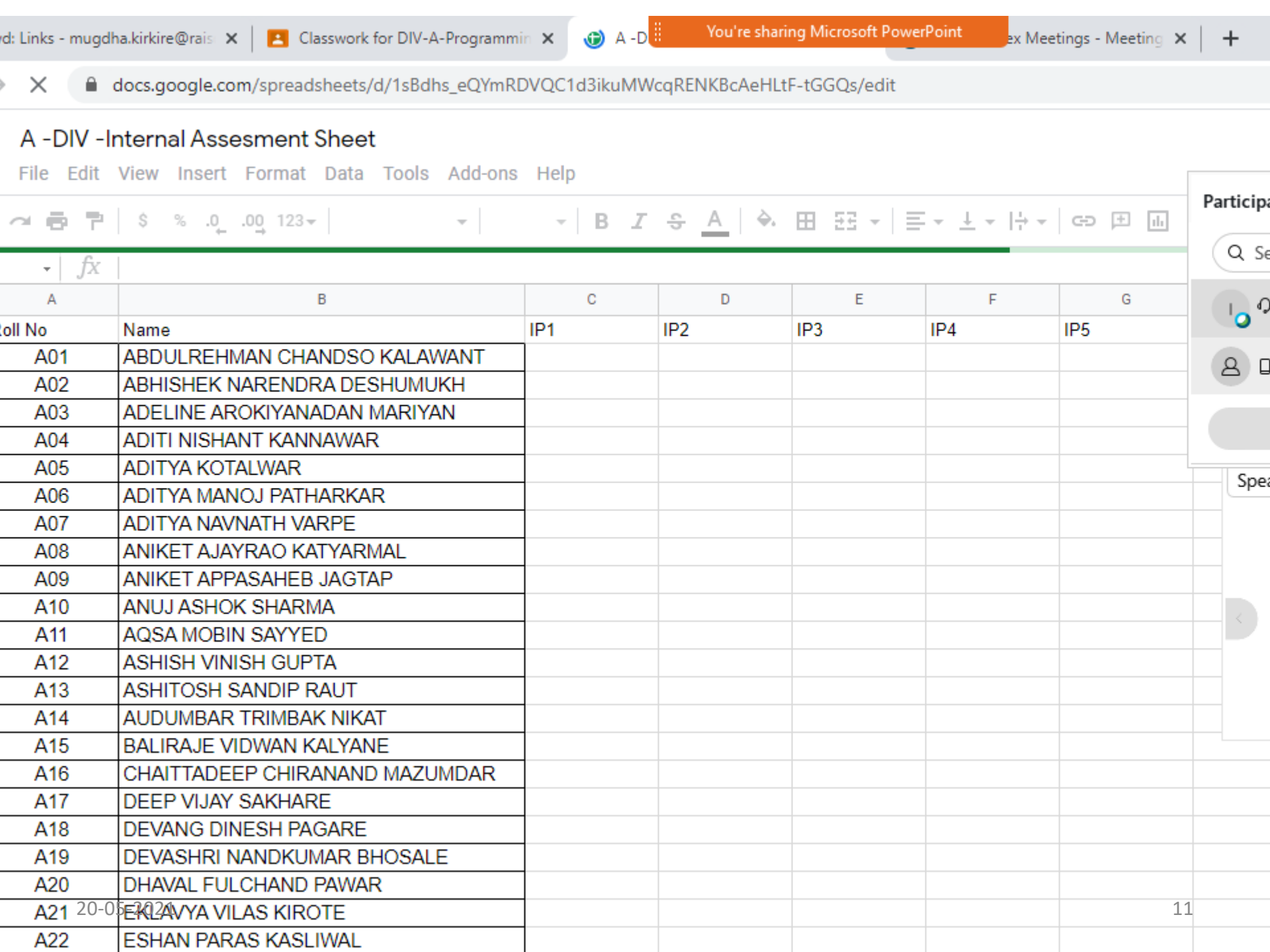
- https://www.udemy.com/
- https://www.coursera.org/
- https://www.cybrary.it/info/freeittraining/
- https://nptel.ac.in/

# Programming?

- What is Programming?
- Why Programming?
- Problem?
- What are different programming languages?
- Why python?

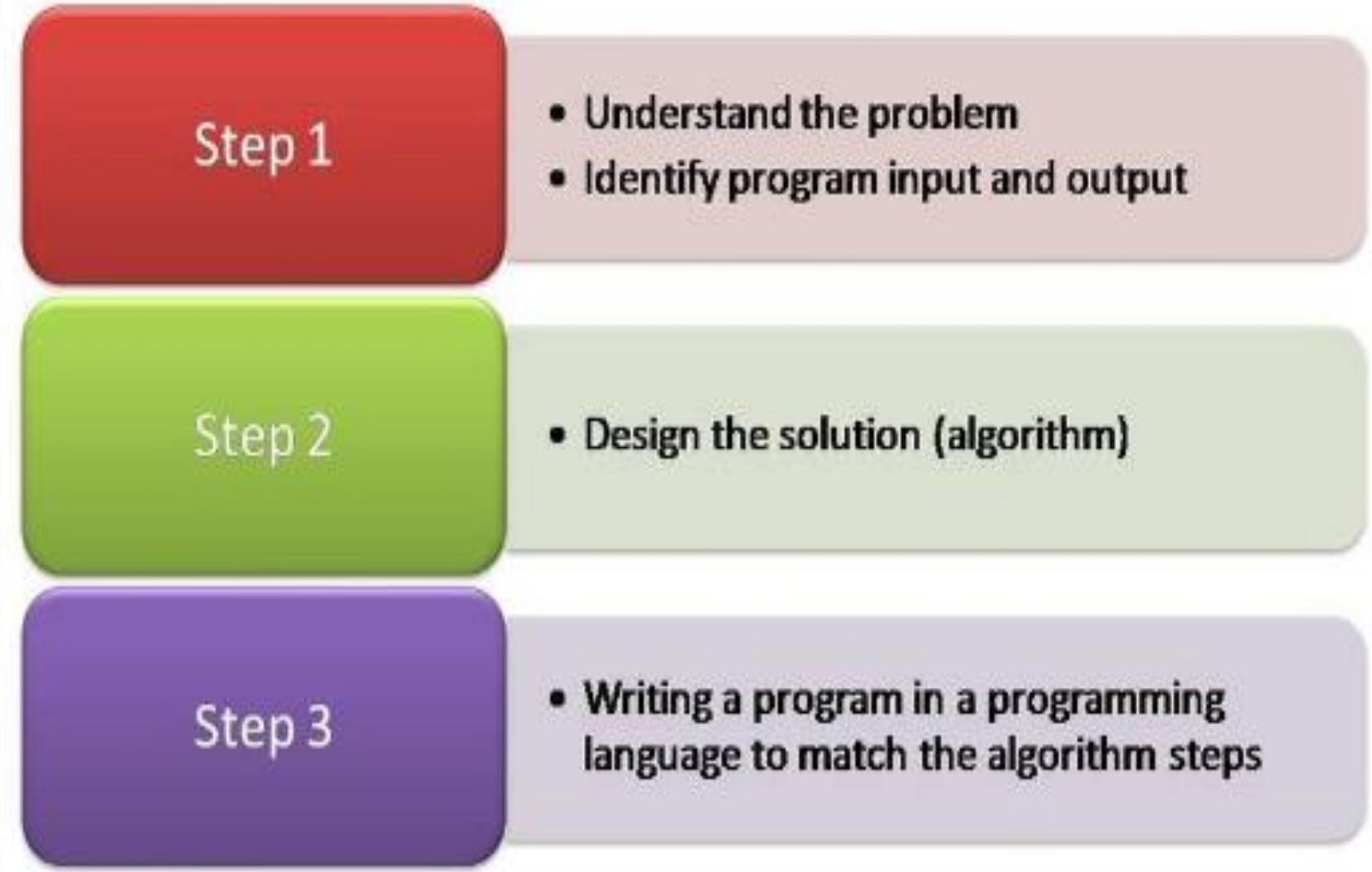
# What is programming

- **Programming** is the process of creating a set of instructions that tell a computer how to perform a task. **Programming** can be done using a variety of computer **programming** languages, such as JavaScript, Python, and C++.
- The most valuable part of learning to program is learning how to think about **arranging the sequence of instructions** to solve the problem or carry out the task




| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| oll No | Name | IP1 | IP2 | IP3 | IP4 | IP5 |
| A01 | ABDULREHMAN CHANDSO KALAWANT | | | | | |
| A02 | ABHISHEK NARENDRA DESHUMUKH | | | | | |
| A03 | ADELINE AROKIYANADAN MARIYAN | | | | | |
| A04 | ADITI NISHANT KANNAWAR | | | | | |
| A05 | ADITYA KOTALWAR | | | | | |
| A06 | ADITYA MANOJ PATHARKAR | | | | | |
| A07 | ADITYA NAVNATH VARPE | | | | | |
| A08 | ANIKET AJAYRAO KATYARMAL | | | | | |
| A09 | ANIKET APPASAHEB JAGTAP | | | | | |
| A10 | ANUJ ASHOK SHARMA | | | | | |
| A11 | AQSA MOBIN SAYYED | | | | | |
| A12 | ASHISH VINISH GUPTA | | | | | |
| A13 | ASHITOSH SANDIP RAUT | | | | | |
| A14 | AUDUMBAR TRIMBAK NIKAT | | | | | |
| A15 | BALIRAJE VIDWAN KALYANE | | | | | |
| A16 | CHAITTADEEP CHIRANAND MAZUMDAR | | | | | |
| A17 | DEEP VIJAY SAKHARE | | | | | |
| A18 | DEVANG DINESH PAGARE | | | | | |
| A19 | DEVASHRI NANDKUMAR BHOSALE | | | | | |
| A20 | DHAVAL FULCHAND PAWAR | | | | | |
| A21 | EKLAVYA VILAS KIROTE | | | | | |
| A22 | ESHAN PARAS KASLIWAL | | | | | |

# Problem Solving



# Programming for Problem Solving

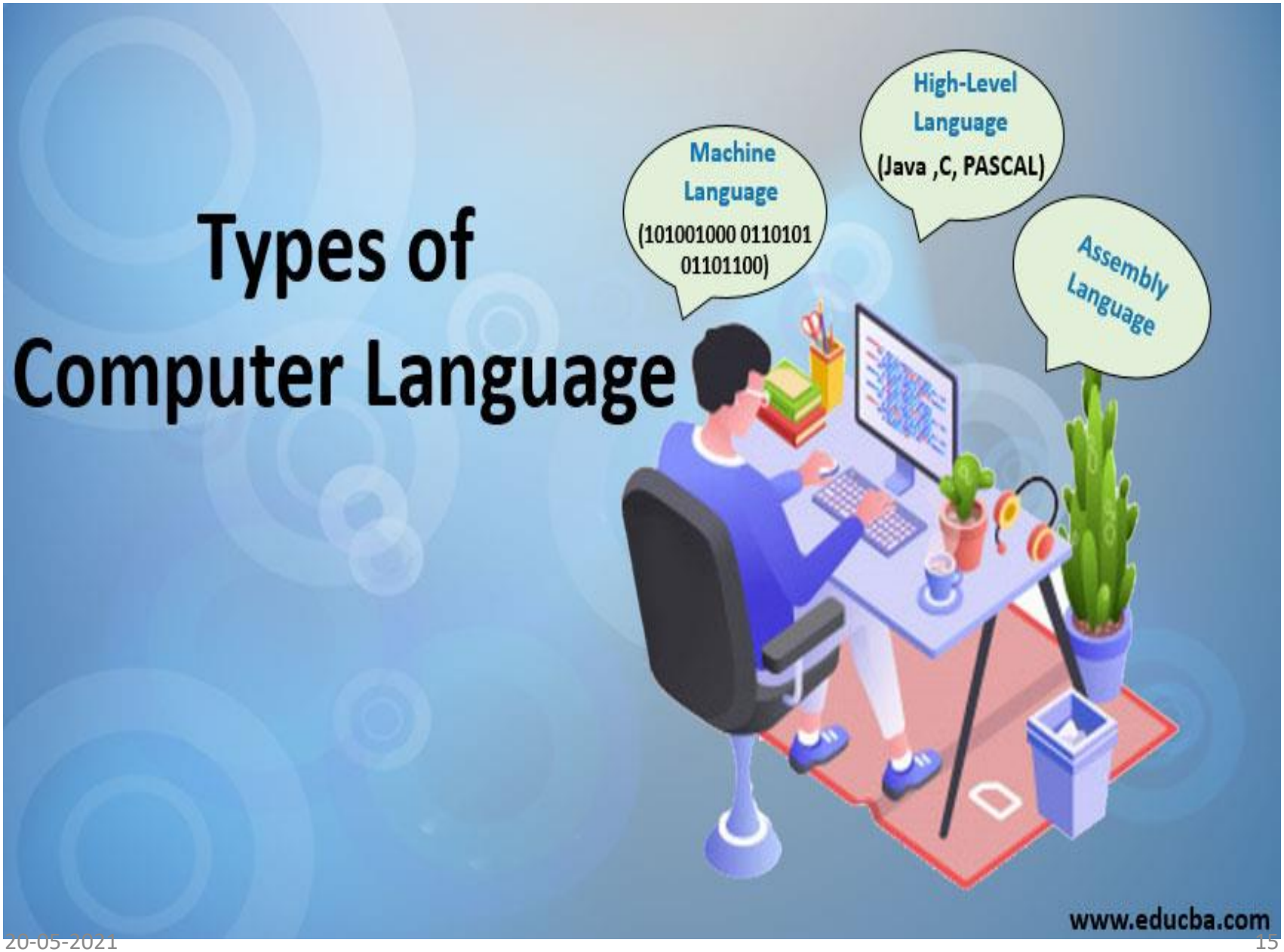
- The process of **defining a problem**, searching for relevant information and resources about the problem, and of **discovering**, **designing**, and **evaluating** the solutions for further opportunities. Includes:

    1. Finding an Answer to a Question
    2. Figuring out how to Perform a Task
    3. Figure out how to Make Things Work

- Not enough to know a particular programming language… Must be able to problem solve

# Programming Languages

- Computers can not use human languages, and programming in the binary language of computers is a very difficult, tedious process.
- Therefore, most programs are written using a programming language and are converted to the binary language used by the computer
- Three major categories of programming languages :
  - Machine Language
  - Assembly Language
  - High level Language


Types of
Computer Language
Machine
Language
(101001000 0110101
01101100)
High-Level
Language
(Java ,C, PASCAL)
Assembly
Language
www.educba.com

# Machine Language

- Natural language of a particular computer
- Primitive **instructions** built into every computer
- The instructions are in the form of binary code(01)
- Any other types of languages must be translated down to this level

# Assembly Languages

- English-like **Abbreviations** used for operations (Load R1, R8)
- Assembly languages were developed to make programming easier
- The computer cannot understand assembly language - a program called assembler is used to convert assembly language programs into machine code

# High-Level Languages

- **English-like commands** and easy to learn and program
- Common mathematical notation
  - Total Cost = Price + Tax;
  - area = 5 * 5 * 3.1415;
- Java, C, C++, FORTRAN, VISUAL BASIC,

# What is Python

- **Python**
  - high-level programming language
  - general-purpose interpreted
  - object-oriented
  - interactive

- **History**
  - created by Guido van Rossum during 1985- 1990

# Why Python


Reading Published Script of Comedy Show


Implementing Programming Language

# Why Python


- When Guido van Rossum implementing **Python**, he was also reading the published scripts from “Monty **Python's** Flying Circus”, a BBC comedy series from the 1970s. Van Rossum thought he needed a **name** that was short, unique, and slightly mysterious, so he decided to call the **language Python**.

# Why Python


Reading Published Script of Comedy Show


Implementing Programming Language

# History of Python

- Invented in the Netherlands, early 90s
  by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the beginning
- Increasingly popular

# Python Features

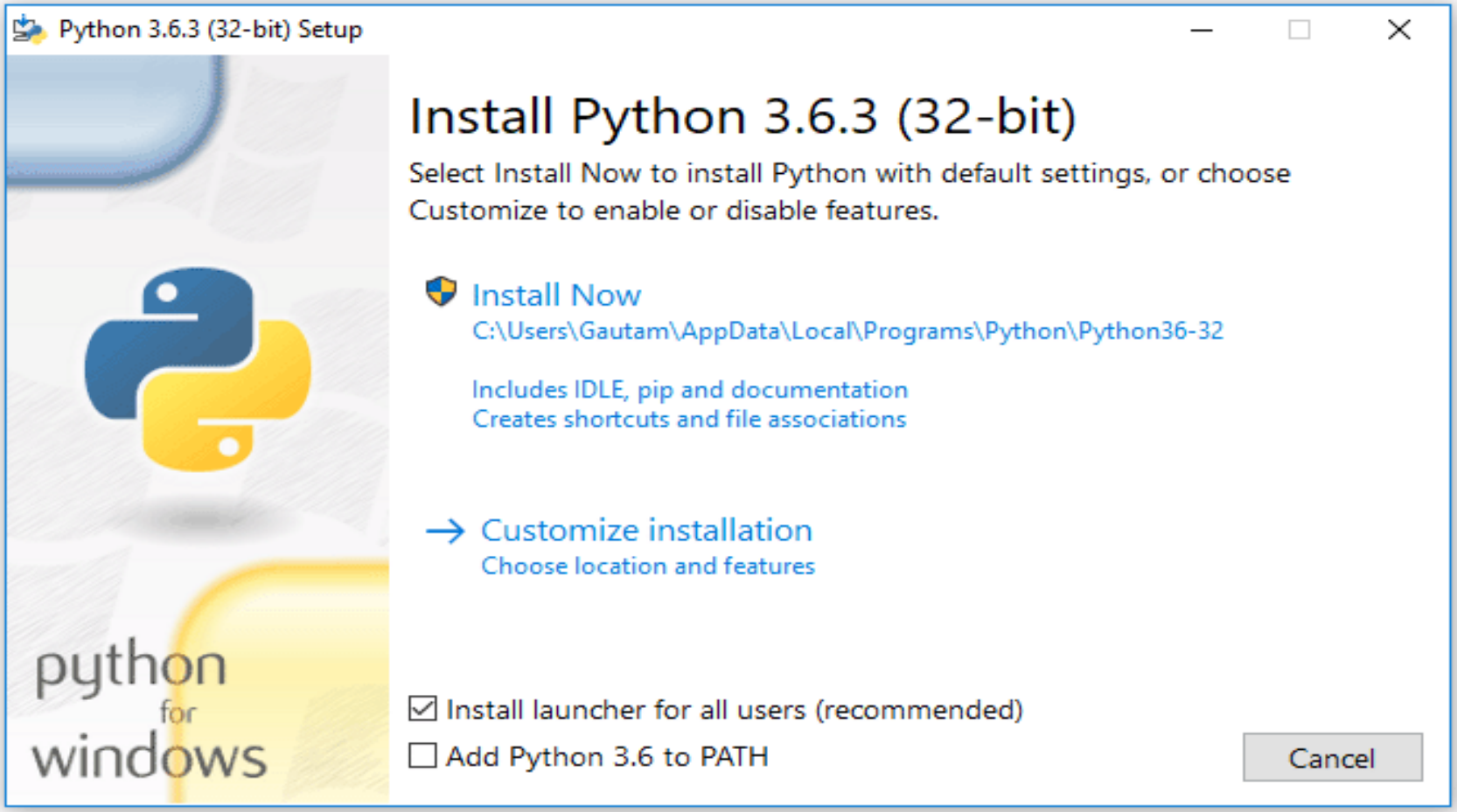
- **Easy to code:**
- **Platform Independent:**
- **Object-Oriented Language:**
- **GUI Programming Support:**
- **High-level Language:**
- **Portable language:**
- **Integrated and Interpreted Language: .**

# How to Install Python

Install Python on Linux :

- The Python interpreter is usually installed as **/usr/local/bin/python** on those machines where it is available; putting: **/usr/local/bin** in your UNIX shell's search path makes it possible to start it by typing the command "**python**" to the shell.
- Python 3 can be installed On Ubuntu Linux by using the following command from the terminal.
- $sudo apt-get install python3-minimal

- Install Python on Windows :
- On Windows machines, a Python installation is usually placed in **C:\Users\UserName\AppData\Local\Programs\Python\Python(Version)**, although you can change this while running the installer.
- To install Python on a Windows machine, follow these steps:
- Download Python for Windows.
- Run the Python installer.
- Here you will see the Python wizard, which is very easy to use. Just accept the default recommended settings and click on the Next button, wait until the install is complete and you are done.


Python 3.6.3 (32-bit) Setup
Install Python 3.6.3 (32-bit)
Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.
Install Now
C:\Users\Gautam\AppData\Local\Programs\Python\Python36-32
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python
for
windows
Install launcher for all users (recommended)
Add Python 3.6 to PATH
Cancel

# Python IDE

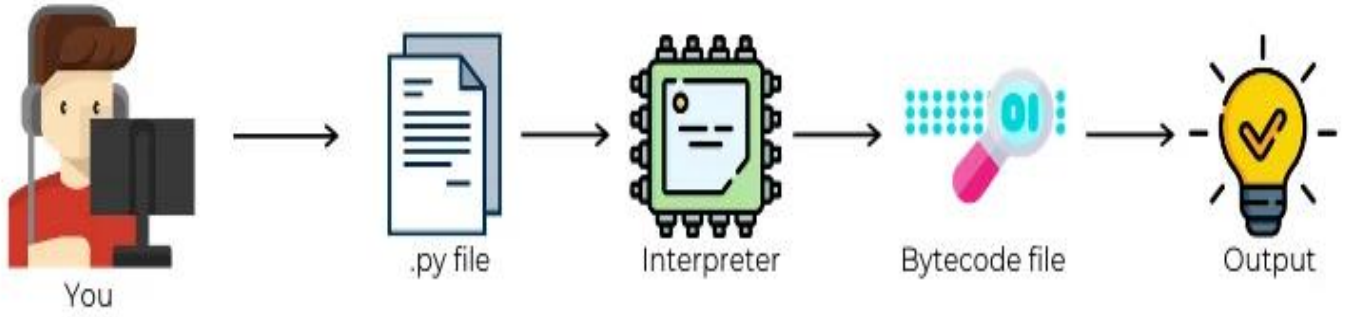
- There are various GUI based Python IDE that python programmers can use for better coding experience. Names of some Python interpreters are:

- PyCharm
- Python IDLE
- The Python Bundle
- pyGUI
- Sublime Text etc.

# Use of Python IDE on Mobile

- There are many android apps like **QPython3, Kivy, BeeWare, pyqtdeploy** etc are available to run python scripts.

# What is Python?




## The process to generate the output:

>> Write a high-level python code.

>> Save the code in .py file.

>> Interprete the code.

>> It will generate a bytecode file.

>>The output will get printed on the screen.

PrepInsta