

Revision

This is an excellent Level 1 completed syllabus. These 20 programs gradually introduce you to input, output, variables, arithmetic operators, and basic mathematical formulas. Below are **complete Python programs** that are easy for beginners to practice again in the lab .

Level 1 – Basic Python (Programs 1–20)

GK InfoTech Solutions –Institute of Engineers

Program 1 – Print "Hello, Python!"

Aim

Print a simple message on the screen.

```
print("Hello, Python!")
print("Welcome to GK InfoTech Solutions")
```

Program 2 – Print Your Name, Course, and College

```
print("Name      : Raju")
print("Course   : Python Programming")
print("College  : ABC Engineering College")
```

Program 3 – Display a Formatted Student ID Card

```
print("=====")
print("      STUDENT ID CARD")
print("=====")
print("Name      : Raju")
print("Roll No   : 101")
print("Course    : Python")
print("College   : ABC Engineering College")
print("=====")
```

Program 4 – Accept and Display Student Details

```
name = input("Enter Student Name : ")
roll = input("Enter Roll Number : ")
course = input("Enter Course : ")

print("\n----- STUDENT DETAILS -----")
print("Name      :", name)
print("Roll      :", roll)
print("Course    :", course)
```

Program 5 – Add Two Numbers

```
num1 = int(input("Enter First Number : "))
num2 = int(input("Enter Second Number : "))

sum = num1 + num2

print("Addition =", sum)
```

Program 6 – Subtract Two Numbers

```
num1 = int(input("Enter First Number : "))
num2 = int(input("Enter Second Number : "))

difference = num1 - num2

print("Subtraction =", difference)
```

Program 7 – Multiply Two Numbers

```
num1 = int(input("Enter First Number : "))
num2 = int(input("Enter Second Number : "))

product = num1 * num2

print("Multiplication =", product)
```

Program 8 – Divide Two Numbers

```
num1 = float(input("Enter First Number : "))
num2 = float(input("Enter Second Number : "))

division = num1 / num2

print("Division =", division)
```

Program 9 – Find the Remainder Using %

```
num1 = int(input("Enter First Number : "))
num2 = int(input("Enter Second Number : "))

remainder = num1 % num2

print("Remainder =", remainder)
```

Program 10 – Calculate the Average of Three Numbers

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))
c = int(input("Enter Third Number : "))

average = (a + b + c) / 3

print("Average =", average)
```

Program 11 – Swap Two Numbers (Using a Temporary Variable)

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))

print("\nBefore Swapping")
print("a =", a)
print("b =", b)

temp = a
a = b
b = temp

print("\nAfter Swapping")
print("a =", a)
print("b =", b)
```

Program 12 – Swap Two Numbers (Without a Temporary Variable)

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))

print("\nBefore Swapping")
print("a =", a)
print("b =", b)

a, b = b, a

print("\nAfter Swapping")
print("a =", a)
print("b =", b)
```

Program 13 – Calculate the Area of a Rectangle

```
length = float(input("Enter Length : "))  
width = float(input("Enter Width : "))  
  
area = length * width  
  
print("Area of Rectangle =", area)
```

Program 14 – Calculate the Area of a Circle

```
radius = float(input("Enter Radius : "))  
  
pi = 3.14159  
  
area = pi * radius * radius  
  
print("Area of Circle =", area)
```

Program 15 – Convert Celsius to Fahrenheit

Formula:

$$\text{Fahrenheit} = (\text{Celsius} \times 9/5) + 32$$

```
celsius = float(input("Enter Temperature in Celsius : "))  
  
fahrenheit = (celsius * 9/5) + 32  
  
print("Temperature in Fahrenheit =", fahrenheit)
```

Program 16 – Convert Fahrenheit to Celsius

Formula:

$$\text{Celsius} = (\text{Fahrenheit} - 32) \times 5/9$$

```
fahrenheit = float(input("Enter Temperature in Fahrenheit : "))  
  
celsius = (fahrenheit - 32) * 5/9  
  
print("Temperature in Celsius =", celsius)
```

Program 17 – Calculate Simple Interest

Formula:

$$SI = (P \times R \times T) / 100$$

```
principal = float(input("Enter Principal Amount : "))
rate = float(input("Enter Rate of Interest : "))
time = float(input("Enter Time (Years) : "))

simple_interest = (principal * rate * time) / 100

print("Simple Interest =", simple_interest)
```

Program 18 – Calculate Compound Interest

Formula:

$$CI = P \times (1 + R/100)^T - P$$

```
principal = float(input("Enter Principal Amount : "))
rate = float(input("Enter Annual Interest Rate (%) : "))
time = float(input("Enter Time (Years) : "))

amount = principal * (1 + rate / 100) ** time
compound_interest = amount - principal

print("Amount =", round(amount, 2))
print("Compound Interest =", round(compound_interest, 2))
```

Program 19 – Calculate the Square and Cube of a Number

```
number = int(input("Enter a Number : "))

square = number ** 2
cube = number ** 3

print("Square =", square)
print("Cube =", cube)
```

Program 20 – Convert Seconds into Hours, Minutes, and Seconds

```
seconds = int(input("Enter Total Seconds : "))

hours = seconds // 3600
remaining_seconds = seconds % 3600

minutes = remaining_seconds // 60
seconds = remaining_seconds % 60

print("Hours    :", hours)
print("Minutes  :", minutes)
print("Seconds  :", seconds)
```

Learning Outcomes

After completing these 20 programs, you will be able to:

-  Write and execute simple Python programs.
-  Use `print()` and `input()` effectively.
-  Declare and use variables.
-  Perform arithmetic calculations.
-  Understand integer (`int`) and decimal (`float`) data types.
-  Apply mathematical formulas in Python.
-  Work with assignment and arithmetic operators.
-  Build confidence before moving to conditional statements and loops.

Level 2 – Decision Making (Programs 21–40)

GK InfoTech Solutions –Institute of Engineers

Program 21 – Even or Odd Number

```
num = int(input("Enter a Number : "))

if num % 2 == 0:
    print(num, "is Even")
else:
    print(num, "is Odd")
```

Program 22 – Positive, Negative, or Zero

```
num = float(input("Enter a Number : "))

if num > 0:
    print("Positive Number")
elif num < 0:
    print("Negative Number")
else:
    print("Zero")
```

Program 23 – Voting Eligibility

```
age = int(input("Enter Your Age : "))

if age >= 18:
    print("Eligible for Voting")
else:
    print("Not Eligible for Voting")
```

Program 24 – Largest of Two Numbers

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))

if a > b:
    print("Largest =", a)
else:
    print("Largest =", b)
```

Program 25 – Largest of Three Numbers

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))
c = int(input("Enter Third Number : "))

if a >= b and a >= c:
    print("Largest =", a)
elif b >= a and b >= c:
    print("Largest =", b)
else:
    print("Largest =", c)
```

Program 26 – Smallest of Three Numbers

```
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))
c = int(input("Enter Third Number : "))

if a <= b and a <= c:
    print("Smallest =", a)
elif b <= a and b <= c:
    print("Smallest =", b)
else:
    print("Smallest =", c)
```

Program 27 – Pass or Fail

```
marks = int(input("Enter Marks : "))

if marks >= 35:
    print("Result : PASS")
else:
    print("Result : FAIL")
```

Program 28 – Grade Calculator

```
marks = int(input("Enter Marks : "))

if marks >= 90:
    print("Grade : A+")
elif marks >= 75:
    print("Grade : A")
elif marks >= 60:
    print("Grade : B")
elif marks >= 35:
    print("Grade : C")
else:
    print("Grade : FAIL")
```

Program 29 – Leap Year Checker

```
year = int(input("Enter Year : "))

if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
    print(year, "is a Leap Year")
else:
    print(year, "is Not a Leap Year")
```


Program 30 – Divisible by 5 and 11

```
num = int(input("Enter Number : "))

if num % 5 == 0 and num % 11 == 0:
    print("Divisible by 5 and 11")
else:
    print("Not Divisible by 5 and 11")
```

Program 31 – Character is Uppercase or Lowercase

```
ch = input("Enter a Character : ")

if ch.isupper():
    print("Uppercase Letter")
elif ch.islower():
    print("Lowercase Letter")
else:
    print("Not an Alphabet")
```

Program 32 – Alphabet or Digit

```
ch = input("Enter a Character : ")

if ch.isalpha():
    print("Alphabet")
elif ch.isdigit():
    print("Digit")
else:
    print("Special Character")
```

Program 33 – Vowel or Consonant

```
ch = input("Enter an Alphabet : ").lower()

if ch in "aeiou":
    print("Vowel")
else:
    print("Consonant")
```

Program 34 – Number is Divisible by 2 and 3

```
num = int(input("Enter Number : "))

if num % 2 == 0 and num % 3 == 0:
    print("Divisible by both 2 and 3")
else:
```

```
print("Not Divisible by both")
```

Program 35 – Century Year Checker

```
year = int(input("Enter Year : "))  
  
if year % 100 == 0:  
    print(year, "is a Century Year")  
else:  
    print(year, "is Not a Century Year")
```

Program 36 – Electricity Bill Calculation

```
units = int(input("Enter Units Consumed : "))  
  
if units <= 100:  
    bill = units * 2  
elif units <= 200:  
    bill = (100 * 2) + ((units - 100) * 3)  
else:  
    bill = (100 * 2) + (100 * 3) + ((units - 200) * 5)  
  
print("Electricity Bill = ₹", bill)
```

Program 37 – Income Tax Calculator (Simple Slabs)

```
income = float(input("Enter Annual Income : "))  
  
if income <= 300000:  
    tax = 0  
elif income <= 600000:  
    tax = income * 0.05  
elif income <= 900000:  
    tax = income * 0.10  
else:  
    tax = income * 0.20  
  
print("Income Tax = ₹", tax)
```

Note: These are simplified slabs for learning purposes. Real tax rules change over time.

Program 38 – BMI Calculator

```
weight = float(input("Enter Weight (kg) : "))
height = float(input("Enter Height (m) : "))

bmi = weight / (height ** 2)

print("BMI =", round(bmi, 2))

if bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal Weight")
elif bmi < 30:
    print("Overweight")
else:
    print("Obese")
```

Program 39 – ATM Withdrawal Eligibility

```
balance = float(input("Enter Account Balance : "))
amount = float(input("Enter Withdrawal Amount : "))

if amount <= balance:
    print("Transaction Successful")
    print("Remaining Balance =", balance - amount)
else:
    print("Insufficient Balance")
```

Program 40 – Login with Username and Password

```
username = input("Enter Username : ")
password = input("Enter Password : ")

if username == "admin" and password == "python123":
    print("Login Successful")
else:
    print("Invalid Username or Password")
```

Learning Outcomes

After completing **Level 2**, you will be able to:

-  Use `if`, `if...else`, and `elif` statements.
-  Apply comparison operators (`==`, `!=`, `>`, `<`, `>=`, `<=`).

- ☒ Use logical operators (and, or, not).
- ☒ Solve real-world decision-making problems.
- ☒ Develop logical thinking for programming.

Level 3 – Loops (Programs 41–60)

GK InfoTech Solutions –Institute of Engineers

Program 41 – Print Numbers from 1 to 10

```
for i in range(1, 11):
    print(i)
```

Program 42 – Print Numbers from 10 to 1

```
for i in range(10, 0, -1):
    print(i)
```

Program 43 – Print Even Numbers from 1 to 100

```
for i in range(2, 101, 2):
    print(i)
```

Program 44 – Print Odd Numbers from 1 to 100

```
for i in range(1, 101, 2):
    print(i)
```

Program 45 – Multiplication Table

```
num = int(input("Enter a Number : "))

for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

Program 46 – Sum of First N Numbers

```
n = int(input("Enter a Number : "))

total = 0

for i in range(1, n + 1):
    total += i

print("Sum =", total)
```

Program 47 – Factorial of a Number

```
num = int(input("Enter a Number : "))  
  
fact = 1  
  
for i in range(1, num + 1):  
    fact *= i  
  
print("Factorial =", fact)
```

Program 48 – Reverse Multiplication Table

```
num = int(input("Enter a Number : "))  
  
for i in range(10, 0, -1):  
    print(f"{num} x {i} = {num * i}")
```

Program 49 – Sum of Even Numbers

```
n = int(input("Enter Limit : "))  
  
total = 0  
  
for i in range(2, n + 1, 2):  
    total += i  
  
print("Sum of Even Numbers =", total)
```

Program 50 – Sum of Odd Numbers

```
n = int(input("Enter Limit : "))  
  
total = 0  
  
for i in range(1, n + 1, 2):  
    total += i  
  
print("Sum of Odd Numbers =", total)
```

Program 51 – Reverse a Number

```
num = int(input("Enter Number : "))

reverse = 0

while num > 0:
    digit = num % 10
    reverse = reverse * 10 + digit
    num = num // 10

print("Reversed Number =", reverse)
```

Program 52 – Count Digits

```
num = int(input("Enter Number : "))

count = 0

while num > 0:
    count += 1
    num = num // 10

print("Total Digits =", count)
```

Program 53 – Sum of Digits

```
num = int(input("Enter Number : "))

total = 0

while num > 0:
    digit = num % 10
    total += digit
    num = num // 10

print("Sum of Digits =", total)
```

Program 54 – Palindrome Number

```
num = int(input("Enter Number : "))

original = num
reverse = 0

while num > 0:
    digit = num % 10
    reverse = reverse * 10 + digit
    num = num // 10
```

```
if original == reverse:
    print("Palindrome Number")
else:
    print("Not a Palindrome")
```

Program 55 – Armstrong Number

```
num = int(input("Enter Number : "))

original = num
total = 0

while num > 0:
    digit = num % 10
    total += digit ** 3
    num = num // 10

if original == total:
    print("Armstrong Number")
else:
    print("Not an Armstrong Number")
```

Note: This version is for learning with **3-digit Armstrong numbers** (e.g., 153, 370, 371, 407).

Program 56 – Prime Number

```
num = int(input("Enter Number : "))

count = 0

for i in range(1, num + 1):
    if num % i == 0:
        count += 1

if count == 2:
    print("Prime Number")
else:
    print("Not a Prime Number")
```

Program 57 – Print Prime Numbers from 1 to N

```
n = int(input("Enter Limit : "))

for num in range(2, n + 1):

    count = 0

    for i in range(1, num + 1):
        if num % i == 0:
            count += 1
```

```
if count == 2:  
    print(num)
```

Program 58 – Fibonacci Series

```
terms = int(input("Enter Number of Terms : "))  
  
a = 0  
b = 1  
  
print("Fibonacci Series:")  
  
for i in range(terms):  
    print(a, end=" ")  
    c = a + b  
    a = b  
    b = c
```

Program 59 – Greatest Common Divisor (GCD)

```
num1 = int(input("Enter First Number : "))  
num2 = int(input("Enter Second Number : "))  
  
gcd = 1  
  
for i in range(1, min(num1, num2) + 1):  
    if num1 % i == 0 and num2 % i == 0:  
        gcd = i  
  
print("GCD =", gcd)
```

Program 60 – Least Common Multiple (LCM)

```
num1 = int(input("Enter First Number : "))  
num2 = int(input("Enter Second Number : "))  
  
largest = max(num1, num2)  
  
while True:  
    if largest % num1 == 0 and largest % num2 == 0:  
        print("LCM =", largest)  
        break  
    largest += 1
```

Bonus Loop Programs (Highly Recommended)

After completing Programs 41–60, you should also practice these to strengthen their loop concepts:

61. Print a right-angle star pattern (*)
62. Print an inverted star pattern
63. Print a number triangle
64. Print Floyd's Triangle
65. Print Pascal's Triangle (advanced)
66. Display the multiplication tables from 1 to 10
67. Count even and odd numbers from 1 to N
68. Print all perfect numbers up to N
69. Check whether a number is a perfect number
70. Generate a simple number pyramid

Excellent! This is the right stage to move you from learning **individual data structures** to solving **real-world problems**. Below is  **Level 5 – Lists, Tuples, Sets & Dictionaries (Programs 76–95)** designed as **lab practical's**. Each program focuses on one concept while reinforcing previous topics like loops, conditions, functions, and user input.

Level 5 – Lists, Tuples, Sets & Dictionaries (Programs 76–95)

GK InfoTech Solutions –Institute of Engineers

Program 76 – Create and Display a Student List

```
you = ["Raju", "Kiran", "Anitha", "Sai"]

print("Student List")

for student in you:
    print(student)
```

Program 77 – Add a Student to the List

```
you = ["Raju", "Kiran", "Anitha"]

new_student = input("Enter New Student Name : ")

you.append(new_student)
```

```
print("\nUpdated Student List")
```

```
for student in you:
    print(student)
```

Program 78 – Remove a Student from the List

```
you = ["Raju", "Kiran", "Anitha", "Sai"]

name = input("Enter Student Name to Remove : ")

if name in you:
    you.remove(name)
    print("Student Removed Successfully")
else:
    print("Student Not Found")

print("\nUpdated List")

for student in you:
    print(student)
```

Program 79 – Search a Student in the List

```
you = ["Raju", "Kiran", "Anitha", "Sai"]

name = input("Enter Student Name : ")

if name in you:
    print(name, "Found")
else:
    print(name, "Not Found")
```

Program 80 – Student Marks Processing

```
marks = []

n = int(input("How Many You? : "))

for i in range(n):
    mark = int(input(f"Enter Marks of Student {i+1} : "))
    marks.append(mark)

print("\nStudent Marks")

for mark in marks:
    print(mark)

print("Highest Marks =", max(marks))
print("Lowest Marks =", min(marks))
```

```
print("Average Marks =", sum(marks)/len(marks))
```

Program 81 – Shopping Cart

```
cart = []

while True:

    item = input("Enter Item (type 'done' to finish) : ")

    if item.lower() == "done":
        break

    cart.append(item)

print("\nShopping Cart")

for item in cart:
    print("-", item)
```

Program 82 – Shopping Cart Bill

```
prices = []

n = int(input("How Many Items? : "))

for i in range(n):
    price = float(input(f"Price of Item {i+1} : "))
    prices.append(price)

print("Total Bill = ₹", sum(prices))
```

Program 83 – Inventory Management

```
inventory = ["Keyboard", "Mouse", "Monitor"]

print("Available Products")

for item in inventory:
    print(item)

new_item = input("\nAdd New Product : ")

inventory.append(new_item)

print("\nUpdated Inventory")
```

```
for item in inventory:  
    print(item)
```

Program 84 – Attendance System

```
attendance = []  
  
n = int(input("Number of You : "))  
  
for i in range(n):  
    status = input(f"Student {i+1} (P/A) : ")  
    attendance.append(status.upper())  
  
present = attendance.count("P")  
absent = attendance.count("A")  
  
print("\nPresent =", present)  
print("Absent =", absent)
```

Program 85 – Employee Details

```
employees = []  
  
n = int(input("Number of Employees : "))  
  
for i in range(n):  
    name = input("Employee Name : ")  
    employees.append(name)  
  
print("\nEmployee List")  
  
for emp in employees:  
    print(emp)
```

Program 86 – Contact Book

```
contacts = {}  
  
n = int(input("Number of Contacts : "))  
  
for i in range(n):  
    name = input("Name : ")
```

```
mobile = input("Mobile : ")

contacts[name] = mobile

print("\nContact Book")

for name, mobile in contacts.items():
    print(name, ":", mobile)
```

Program 87 – Search Contact

```
contacts = {
    "Raju": "9876543210",
    "Kiran": "9123456789",
    "Anitha": "9988776655"
}

name = input("Enter Contact Name : ")

if name in contacts:
    print("Mobile Number :", contacts[name])
else:
    print("Contact Not Found")
```

Program 88 – Tuple Example

```
subjects = ("Python", "Java", "C", "HTML")

print("Subjects")

for subject in subjects:
    print(subject)

print("Total Subjects =", len(subjects))
```

Program 89 – Student Roll Numbers Using Tuple

```
roll_numbers = (101,102,103,104,105)

print("Student Roll Numbers")

for roll in roll_numbers:
    print(roll)
```

Program 90 – Remove Duplicate Numbers Using Set

```
numbers = [10,20,10,30,40,20,50]

unique = set(numbers)

print("Original List")
```

```
print(numbers)

print("\nUnique Numbers")

print(unique)
```

Program 91 – Union of Two Sets

```
set1 = {10,20,30,40}

set2 = {30,40,50,60}

print("Union")

print(set1.union(set2))
```

Program 92 – Intersection of Two Sets

```
set1 = {10,20,30,40}

set2 = {30,40,50,60}

print("Intersection")

print(set1.intersection(set2))
```

Program 93 – Student Dictionary

```
student = {

    "Name": "Raju",
    "Roll": 101,
    "Course": "Python",
    "Marks": 85

}

for key, value in student.items():

    print(key, ":", value)
```

Program 94 – Student Result Dictionary

```
student = {

    "Name": "Raju",
    "Python": 80,
    "Java": 75,
    "C": 70

}
```

```
}

total = student["Python"] + student["Java"] + student["C"]

average = total/3

print("Total =", total)
print("Average =", average)
```

Program 95 – Employee Database

```
employees = {

    101:"Raju",
    102:"Kiran",
    103:"Anitha"

}

empid = int(input("Enter Employee ID : "))

if empid in employees:
    print("Employee Name :", employees[empid])
else:
    print("Employee Not Found")
```

Learning Outcomes

After completing **Level 5**, you will be able to:

-  Create and manipulate **lists, tuples, sets, and dictionaries**.
-  Store and process collections of real-world data such as student records, employee details, contacts, attendance, and shopping items.
-  Apply loops, conditions, and built-in functions (`append()`, `remove()`, `max()`, `min()`, `sum()`, `count()`, `union()`, `intersection()`, etc.).
-  Build the foundation needed for **mini projects** such as Student Management Systems, Inventory Systems, and Contact Books.

Level 6 – Intermediate Python (Programs 91–100)

GK InfoTech Solutions –Institute of Engineers

Program 91 – Calculator Using Functions

```
# Calculator Using Functions

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    return a / b

num1 = float(input("Enter First Number : "))
num2 = float(input("Enter Second Number : "))

print("\n1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")

choice = int(input("Choose Operation : "))

if choice == 1:
    print("Result =", add(num1, num2))

elif choice == 2:
    print("Result =", subtract(num1, num2))

elif choice == 3:
    print("Result =", multiply(num1, num2))

elif choice == 4:
    print("Result =", divide(num1, num2))

else:
    print("Invalid Choice")
```


Program 92 – Student Result Using Functions

```
# Student Result Using Functions

def total_marks(m1, m2, m3):
    return m1 + m2 + m3

def average(total):
    return total / 3

def grade(avg):

    if avg >= 90:
        return "A+"

    elif avg >= 75:
        return "A"

    elif avg >= 60:
        return "B"

    elif avg >= 35:
        return "C"

    else:
        return "Fail"

name = input("Student Name : ")

m1 = int(input("Python Marks : "))
m2 = int(input("Java Marks : "))
m3 = int(input("C Marks : "))

total = total_marks(m1, m2, m3)
avg = average(total)

print("\nStudent :", name)
print("Total =", total)
print("Average =", round(avg, 2))
print("Grade =", grade(avg))
```

Program 93 – Random Password Generator

```
import random
import string
```

```
length = int(input("Enter Password Length : "))

characters = string.ascii_letters + string.digits + "@#$$%&"

password = ""

for i in range(length):
    password += random.choice(characters)

print("Generated Password :", password)
```

Program 94 – Number Guessing Game

```
import random

secret = random.randint(1,10)

while True:

    guess = int(input("Guess Number (1-10): "))

    if guess == secret:
        print("Congratulations! You Won.")
        break

    elif guess < secret:
        print("Too Small")

    else:
        print("Too Large")
```

Program 95 – Read Data from a File

```
file = open("student.txt","r")

data = file.read()

print(data)

file.close()
```

Create a file named **student.txt** before running this program.

Program 96 – Write Data to a File

```
file = open("student.txt","w")

name = input("Enter Student Name : ")

file.write(name)
```

```
file.close()

print("Data Written Successfully")
```

Program 97 – Append Data to a File

```
file = open("student.txt", "a")

name = input("Enter Another Student Name : ")

file.write("\n" + name)

file.close()

print("Data Appended Successfully")
```

Program 98 – Division Using Exception Handling

```
try:

    a = int(input("Enter First Number : "))
    b = int(input("Enter Second Number : "))

    result = a / b

    print("Division =", result)

except ZeroDivisionError:

    print("Cannot Divide by Zero")

except ValueError:

    print("Please Enter Numbers Only")

finally:

    print("Program Completed")
```

Program 99 – Student Class Using OOP

```
class Student:

    def __init__(self, name, roll, marks):

        self.name = name
        self.roll = roll
        self.marks = marks

    def display(self):

        print("\nStudent Details")
        print("Name :", self.name)
        print("Roll :", self.roll)
        print("Marks :", self.marks)

name = input("Enter Name : ")
roll = input("Enter Roll Number : ")
marks = int(input("Enter Marks : "))

s1 = Student(name, roll, marks)

s1.display()
```

Program 100 – Student Management System (Mini Project)

```
you = {}

while True:

    print("\n===== STUDENT MANAGEMENT SYSTEM =====")
    print("1. Add Student")
    print("2. Search Student")
    print("3. Display All You")
    print("4. Delete Student")
    print("5. Exit")

    choice = int(input("Enter Choice : "))

    if choice == 1:

        roll = input("Roll Number : ")
        name = input("Student Name : ")

        you[roll] = name

        print("Student Added Successfully")
```

GK InfoTech Solutions – Aganampudi

Learn Python – 21 Days (100 Plus Bonus Program)

```
elif choice == 2:

    roll = input("Enter Roll Number : ")

    if roll in you:
        print("Student Name :", you[roll])
    else:
        print("Student Not Found")

elif choice == 3:

    print("\nStudent Records")

    if len(you) == 0:
        print("No Records Available")
    else:
        for roll, name in you.items():
            print(roll, "-", name)

elif choice == 4:

    roll = input("Enter Roll Number to Delete : ")

    if roll in you:
        del you[roll]
        print("Student Deleted Successfully")
    else:
        print("Student Not Found")

elif choice == 5:

    print("Thank You")
    break

else:

    print("Invalid Choice")
```

Learning Outcomes

After completing **Level 6**, you will be able to:

-  Create reusable code using **functions**.
-  Organize programs into smaller, manageable modules.
-  Generate random values using Python libraries.
-  Read, write, and append data using **file handling**.
-  Handle runtime errors using **try-except-finally**.
-  Create **classes** and **objects** using Object-Oriented Programming (OOP).
-  Develop a simple **menu-driven mini project** using dictionaries, loops, conditions, and functions.

Congratulations!

By completing **Levels 1–6**, you will have practiced **100 Python programs**, covering:

-  Python Basics
-  Variables & Data Types
-  Operators
-  Conditional Statements
-  Loops
-  Strings
-  Lists, Tuples, Sets & Dictionaries
-  Functions
-  Modules
-  File Handling
-  Exception Handling
-  Object-Oriented Programming (OOP)
-  Mini Projects

This progression provides a solid foundation for advancing to **Bonus Challenge Programs (101–120)** and future learning in **Data Science, Artificial Intelligence, Machine Learning, Web Development, and Automation**.

Excellent choice. Below is a **beginner-friendly ATM Management System** designed for your Python you. It uses the concepts learned in the 21-day course:

-  Variables
-  Input/Output
-  if...elif...else
-  while Loop
-  Functions
-  Exception Handling
-  Menu-Driven Programming

This is suitable for **Level 6 / Bonus Challenge Program 101**.

Program 101 – ATM Management System

Aim

Develop a simple ATM Management System using Python.

Concepts Used

- Functions
- Variables
- Loops
- Conditional Statements
- Exception Handling

Algorithm

Start

↓

Enter ATM PIN

↓

If PIN Correct

↓

Display ATM Menu

1.Check Balance

2.Deposit

3.Withdraw

4.Mini Statement

5.Change PIN

6.Exit

↓

Perform Selected Operation

↓

Prepared by Gk;

Repeat Until Exit

↓

Stop

Python Program

```
# =====
#           ATM MANAGEMENT SYSTEM
#       GK InfoTech Solutions - Aganampudi
# =====

balance = 10000
pin = "1234"

transactions = []

# -----
# Check Balance
# -----
def check_balance():
    print("\nCurrent Balance = ₹", balance)

# -----
# Deposit Money
# -----
def deposit():

    global balance

    amount = float(input("Enter Deposit Amount : ₹"))

    if amount <= 0:
        print("Invalid Amount")
        return

    balance += amount

    transactions.append(f"Deposited : ₹{amount}")

    print("Amount Deposited Successfully")

# -----
# Withdraw Money
# -----
def withdraw():

    global balance

    amount = float(input("Enter Withdraw Amount : ₹"))

    if amount <= 0:
        print("Invalid Amount")

    elif amount > balance:
```


GK InfoTech Solutions – Aganampudi

Learn Python – 21 Days (100 Plus Bonus Program)

```

print("Insufficient Balance")

else:

    balance -= amount

    transactions.append(f"Withdrawn : ₹{amount}")

    print("Please Collect Your Cash")

# -----
# Mini Statement
# -----
def mini_statement():

    print("\n===== MINI STATEMENT =====")

    print("Current Balance : ₹", balance)

    if len(transactions) == 0:

        print("No Transactions Available")

    else:

        for t in transactions:
            print(t)

# -----
# Change PIN
# -----
def change_pin():

    global pin

    old_pin = input("Enter Old PIN : ")

    if old_pin == pin:

        new_pin = input("Enter New PIN : ")

        confirm = input("Confirm New PIN : ")

        if new_pin == confirm:

            pin = new_pin

            print("PIN Changed Successfully")

        else:

            print("PIN Does Not Match")

    else:

        print("Wrong Old PIN")

```

GK InfoTech Solutions – Aganampudi

Learn Python – 21 Days (100 Plus Bonus Program)

```
# -----
# Main Program
# -----

print("=====")
print("      WELCOME TO PYTHON ATM")
print("=====")

attempt = 0

while attempt < 3:

    entered_pin = input("Enter ATM PIN : ")

    if entered_pin == pin:

        print("\nLogin Successful")

        while True:

            print("\n===== ATM MENU =====")
            print("1. Check Balance")
            print("2. Deposit Money")
            print("3. Withdraw Money")
            print("4. Mini Statement")
            print("5. Change PIN")
            print("6. Exit")

            try:

                choice = int(input("Enter Your Choice : "))

                if choice == 1:
                    check_balance()

                elif choice == 2:
                    deposit()

                elif choice == 3:
                    withdraw()

                elif choice == 4:
                    mini_statement()

                elif choice == 5:
                    change_pin()

                elif choice == 6:

                    print("\nThank You for Using Python ATM")
```

GK InfoTech Solutions – Aganampudi

Learn Python – 21 Days (100 Plus Bonus Program)

```

        print("Visit Again")

        break

    else:

        print("Invalid Choice")

    except ValueError:

        print("Please Enter Numbers Only")

        break

else:

    attempt += 1

    print("Wrong PIN")

    print("Attempts Left :", 3 - attempt)

if attempt == 3:

    print("\nATM Card Blocked")
    print("Contact Your Bank")

```

Sample Output

```

=====
      WELCOME TO PYTHON ATM
=====

Enter ATM PIN : 1234

Login Successful

===== ATM MENU =====
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Mini Statement
5. Change PIN
6. Exit

Enter Your Choice : 2

Enter Deposit Amount : ₹5000

Amount Deposited Successfully

```

Example Mini Statement

```

===== MINI STATEMENT =====

```

Current Balance : ₹14500

Deposited : ₹5000

Withdrawn : ₹500

Lab Exercise

Modify the program to:

- Add Customer Name
 - Add Account Number
 - Display Date & Time
 - Show Last 5 Transactions
 - Limit Withdrawal to ₹20,000 per day
-

Challenge Exercise

Enhance the ATM by adding:

- Fast Cash (₹500, ₹1000, ₹2000)
 - Balance Transfer
 - Change Language
 - PIN must contain exactly 4 digits
 - Save transactions to a file (`transactions.txt`)
-

Learning Outcomes

After completing this practical, you will be able to:

- ☒ Design a menu-driven application.
 - ☒ Use functions to organize code.
 - ☒ Manage data using variables and lists.
 - ☒ Apply loops and conditional statements.
 - ☒ Handle user input errors with exception handling.
 - ☒ Develop the foundation for larger real-world projects.
-

GK InfoTech Solutions – Aganampudi
Learn Python – 21 Days (100 Plus Bonus Program)

This project is intentionally written for beginners. As your you progress, we can create an Advanced ATM Management System (Version 2) using Object-Oriented Programming (OOP), file handling, login accounts, transaction history, and persistent data storage, making it similar to a real banking application.