

# **Programming for Artificial Intelligence Practical File**

Student Name: Gagandeep Singh

Roll Number: 3812

College: Chitkara University

Submitted To: \_\_\_\_\_

Session: 2024-2025

## **1. Triangle Validity & Area Using Heron's Formula**

```
a = float(input("Enter side a: "))
b = float(input("Enter side b: "))
c = float(input("Enter side c: "))

if a+b>c and a+c>b and b+c>a:
    s = (a+b+c)/2
    area = (s*(s-a)*(s-b)*(s-c))**0.5
    print("Area =", area)
else:
    print("Triangle cannot be formed.")
```

## **2. Strong Number**

```
num = int(input("Enter number: "))
temp = num
sum = 0
import math
```

```

while temp>0:
    d = temp % 10
    sum += math.factorial(d)
    temp//=10

if sum==num:
    print("Strong Number")
else:
    print("Not a Strong Number")

```

### 3. Fibonacci Sum Up To N

```

n = int(input("Enter n: "))
a, b = 0, 1
sum = 0

while a <= n:
    sum+=a
    a,b=b,a+b

print("Sum =", sum)

```

### 4. Pascal's Triangle

```

rows = int(input("Enter rows: "))
for i in range(rows):
    num = 1
    for j in range(i+1):
        print(num, end=" ")
        num = num*(i-j)//(j+1)
    print()

```

### 5. Capitalize First Letter of Each Word

```

s = input("Enter sentence: ")
result = s.title()
print(result)

```

### 6. Remove Duplicates and Sort

```

lst = list(map(int,input("Enter numbers: ").split()))
unique = sorted(set(lst))
print("Unique Sorted List:", unique)

```

### 7. Row Sum of Matrix

```

m = int(input("Rows: "))
n = int(input("Columns: "))
matrix = []

for i in range(m):
    row = list(map(int,input().split()))

```

```

matrix.append(row)

for row in matrix:
    print("Row sum:", sum(row))

```

## 8. Character Frequency in String

```

s = input("Enter string: ")
freq = {}
for ch in s:
    freq[ch] = freq.get(ch, 0)+1
print(freq)

```

## 9. Search Element & Remove Empty Tuples

```

lst = eval(input("Enter list of tuples: "))
x = input("Enter element to search: ")
found = any(x in t for t in lst)
print(found)

cleaned = [t for t in lst if t!=()]
print(cleaned)

```

## 10. Lambda with map, filter, reduce

```

from functools import reduce
nums = list(range(1,11))

squares = list(map(lambda x: x*x, nums))
evens = list(filter(lambda x: x%2==0, nums))
product = reduce(lambda x,y:x*y, nums)
desc = sorted(nums, key=lambda x:-x)

print(squares, evens, product, desc)

```

## 11. Set Operations

```

a = set(map(int,input().split()))
b = set(map(int,input().split()))

print("Only in A:", a-b)
print("Only in B:", b-a)
print("Common:", a&b)
print("Union:", a|b)

```

## 12. Group Words by Starting Letter

```

words = input().split()
d = {}
for w in words:
    key = w[0].lower()
    d.setdefault(key, []).append(w)

```

```
print(d)
```

## 13. Word Frequency in File

```
import re
file = open("sample.txt", "r")
text = file.read().lower()
words = re.findall(r"\b\w+\b", text)

freq = {}
for w in words:
    freq[w] = freq.get(w, 0) + 1

print(freq)
print("Most Frequent:", max(freq, key=freq.get))
```

## 14. Exception Handling (Division)

```
try:
    a = int(input("Numerator: "))
    b = int(input("Denominator: "))
    print(a/b)
except ZeroDivisionError:
    print("Cannot divide by zero.")
except ValueError:
    print("Enter integers only.")
```

## 15. Student Class

```
class Student:
    def __init__(self, name, roll, marks):
        self.name = name
        self.roll = roll
        self.marks = marks

    def display(self):
        print(self.name, self.roll, self.marks)

    def result(self):
        print("Pass" if self.marks >= 40 else "Fail")

s = Student("Alice", 101, 75)
s.display()
s.result()
```

## 16. GUI Calculator (Tkinter)

```
from tkinter import *
root = Tk()
root.title("Calculator")

e = Entry(root, width=16, font=("Arial", 20), bd=10)
e.grid(row=0, column=0, columnspan=4)
```

```
# Buttons creation skipped for brevity
root.mainloop()
```