

Infosys Springboard Virtual Internship 6.0: Milestone 1 Report

Project Name : Crypto Volatility and Risk Analyzer

Intern Name : Vyshnavi Bommisetty

Batch : 11 | **Team:** C

Mentor : Sangeetha Mahalingam

EXECUTIVE SUMMARY

This document presents the learning progress and technical activities completed during Milestone–1 of the Infosys Springboard Virtual Internship. The milestone focused on building a strong foundation in Python programming through hands-on practice using Google Colab notebooks. The sessions emphasized console-based execution, data handling, string processing, and logical decision-making, which are essential for advanced project development.

1. LEARNING ENVIRONMENT OVERVIEW

The internship sessions were conducted using Google Colab, a cloud-based Python execution platform. The environment allowed the integration of executable Python code cells along with markdown-based explanations. This setup helped in understanding program logic, executing code interactively, and documenting outcomes efficiently.

2. KNOWLEDGE AREAS ADDRESSED

During Milestone–1, the following learning areas were covered:

- Python program structure and syntax
- Console-based user input and output
- Integer and floating-point operations
- String manipulation techniques
- Use of built-in Python modules
- Conditional and logical operations
- Documentation using markdown cells

3. PROGRAM EXECUTION AND ANALYSIS

This section details the Python programs practiced during the sessions along with their purpose, implementation, and outputs.

Program 1: User Interaction Using Console

Objective:

To understand how Python programs interact with users through console input and output.

Code Snippet:

```
print("Hi..!")  
name = input('Enter your Name: ')  
print('Hi ', name)
```

Output:

```
Hi..!  
Enter your Name: Vyshu  
Hi Vyshu
```

Program 2: Numerical Computations Using Data Types

Objective:

To perform arithmetic operations using integer and floating-point data types.

Explanation:

User inputs are taken from the console. Integer values are added, and floating-point values are used to calculate the area of a triangle.

Code Snippet:

```
# Integer Input  
a = int(input('Enter the value for A: '))  
b = int(input('Enter the value for B: '))  
c = a+b  
print(c)  
  
# Area of Triangle  
b = float(input("Enter the Base Length of Triangle: "))  
h = float(input("Enter the Height Length of Triangle: "))  
a = 0.5 * b * h  
print("Area = ", a)
```

Output:

```
106 Area = 31.5
```

Program 3: Utilizing Built-in Python Libraries

Objective:

To explore Python's built-in modules for practical applications.

Explanation:

The calendar module is used to generate a formatted monthly calendar based on user input.

Code Snippet:

```
import calendar

year = int(input("Please Enter the year Number: "))
month = int(input("Please Enter the month Number: "))

print(calendar.month(year, month))
```

Output:

```
    June 2005
Mo Tu We Th Fr Sa Su
    1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

Program 4a: Text Processing Using String Operations

Objective:

To analyze and manipulate textual data using string methods.

Code Snippet:

```
# Looping through string
for x in 'MISSISSIPPI':
    print(x)

# String Methods
s = "life iS beautiful !!! "
print(s.upper())
print(s.replace("l",'w'))
print(s.split("i"))
```

Output:

```
M
I
S
S
I
S
S
I
P
P
I
LIFE IS BEAUTIFUL !!!
wife iS beautifuw !!!
['l', 'fe ', 'S beaut', 'ful !!! ']
```

4b. String Slicing and Concatenation

I practiced accessing specific parts of strings using slicing techniques on the string "Python Programming". I also demonstrated string concatenation by joining two separate phrases into a single cohesive sentence.

Code Snippet:

```
str = "Python Programming"
# Slicing from 1st to 4th index
print(str[1:5])
# Slicing from 0th to 2nd index
print(str[:3])

a = 'life is beautiful !!!'
b = 'Live happily'
print("Concatenation: " + a + b)
```

Output:

```
ytho
Pyt
Concatenation: life is beautiful !!!Live happily
```

4c. String Formatting

To practice dynamic text generation, I utilized the .format() method. By using positional arguments, I injected variables into a sentence structure.

Code Snippet:

```
# Positional arguments in formatting
print('{1} and {0} are best friends'.format('Vyshu','Sindhu'))
```

Output:

```
Sindhu and Vyshu are best friends
```

Program 5: Decision Making Using Conditional Statements

Objective:

To implement decision-making logic using if-else conditions.

Code Snippet:

```
# Even or Odd Check
num = int(input('Enter the number : '))

if num % 2 == 0:
    print('even: ', num)
else:
    print('odd: ', num)

# Greatest of three numbers
a = int(input('Enter A: '))
b = int(input('Enter B: '))
c = int(input('Enter C: '))

if (c>a and c>b):
    print('C is bigger')
```

Output:

```
odd: 29
C is bigger
```

Program 6: Logical Validation Using Operators

Objective:

To validate conditions using logical operators.

Code Snippet:

```
# Range check
num = int(input("Guess a Number: "))
if num>2 and num<79:
    print("Correct Guess!")
else:
    print("Incorrect Guess!")

# List membership
regno = int(input("Enter Register no : "))
if regno in [98,101,105,109,110]:
    print("pass")
else:
    print("Fail")
```

Output:

Incorrect Guess!

Fail

4. PARTICIPATION AND SKILL DEVELOPMENT

The intern actively participated in daily sessions by writing and executing Python programs. Logical explanations were documented using markdown cells, which improved problem-solving ability, debugging skills, and overall programming confidence.

5. OVERALL OUTCOME

- Strengthened understanding of Python fundamentals
- Improved ability to write and analyze console-based programs
- Gained experience in documenting technical workflows
- Built a solid foundation for future milestones

6. FINAL REMARKS

Milestone–1 successfully established a strong base in Python programming concepts. The hands-on learning approach and continuous practice during the sessions will be highly beneficial for implementing advanced features in upcoming milestones of the project.