Gabriel Q. Escobido

BSCoE – 2A

SOFTWARE DESIGN

Laboratory Exercise No. 2 CH2

Title: Understanding Programming Constructs

Brief Introduction

Programming constructs are the building blocks of any software application. These include variables, loops, conditionals, functions, data structures, and recursion. This lab focuses on these constructs and their implementation in Python.

Objectives

- Learn and implement basic programming constructs.
- Explore recursion with practical examples.

Detailed Discussion

Variables: Variables are containers for storing data values. In Python, variables do not need explicit declaration and can hold data of any type.

Loops: Loops are used to execute a block of code repeatedly. Python supports for and while loops.

Conditional Statements: These allow branching based on conditions. Python uses if, elif, and else for decision-making.

Functions: Functions are reusable blocks of code that perform specific tasks. They can accept inputs (parameters) and return outputs.

Data Structures: Python supports several built-in data structures like lists, dictionaries, sets, and tuples to organize and manage data efficiently.

Recursion: Recursion is a technique where a function calls itself to solve a problem. It is useful for tasks like calculating factorials, generating Fibonacci sequences, and traversing data structures.

Detailed Discussion

Programming Paradigm	Description	Example Applications
Imperative	Focuses on step-by-step instructions.	Low-level programming tasks
Object-Oriented	Organizes code using objects and classes.	GUI applications, simulations
Functional	Emphasizes mathematical functions and immutability.	Data analysis, AI
Declarative	Specifies what to do without describing how to do it.	SQL, configuration files
Event-Driven	Responds to events like clicks, signals, or messages.	GUIs, games
Concurrent	Manages multiple computations at the same time.	Web servers, parallel processing

Materials

- Computer system with Python.
- VS Code IDE.

Time Frame

1.5 hours

Procedure

1. Create a new Python file in VS Code.
2. Implement examples of variables, loops, conditionals, and recursion.

Example Code:

```
# Variables and loops
nums = [1, 2, 3, 4, 5]
for num in nums:
  print(num)
```

```
# Conditional Statements
if len(nums) > 3:
  print("List is large")
else:
  print("List is small")

# Recursion
def factorial(n):
  if n == 0:
     return 1
  return n * factorial(n-1)

print(factorial(5))
```

1. Run the program and observe the results.

Results

```
ASUS TUF/Desktop/software design/prjects/prjt 1/prof/LAB2 SOFTWARE DESIGN"
Hello, World!
The number is: 10
The value of Pi is: 3.14159

For Loop Example:
1
2
3
4
5

Conditional Example:
10 is positive.
PS C:\Users\ASUS TUF\Desktop\software design\prjects\prjt 1\prof>
```

EXPLORER

PROF
- library_env
- library_system
- LAB2 SOFTWARE DESIGN
- LAB4
- project

OUTLINE

project | LAB2 SOFTWARE DESIGN | LAB4

LAB2 SOFTWARE DESIGN > ...

```python
# Variable example
greeting = "Hello, World!"  # String variable
number = 10  # Integer variable
pi_value = 3.14159  # Float variable

# Printing the variables
print(greeting)
print("The number is:", number)
print("The value of Pi is:", pi_value)

# Loop example - Using a for loop to print numbers from 1 to 5
print("\nFor Loop Example:")
for i in range(1, 6):
    print(i)

# Conditionals example - Check if a number is positive, negative, or zero
print("\nConditional Example:")
if number > 0:
    print(f"{number} is positive.")
elif number < 0:
    print(f"{number} is negative.")
else:
    print(f"{number} is zero.")

# Recursion example - Factorial of a number
def factorial(n):
    # Base case: factorial of 0 or 1 is 1
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

```

Record outputs for different test cases.

Follow-Up Questions

1. What is the purpose of using recursion?

ANS: The purpose of using recursion is to solve problems by breaking them down into smaller, more manageable subproblems.

2. How do loops differ from recursion?

ANS: Loops repeatedly execute a block of code, while recursion calls a function within itself to achieve repetition.

3. Explain a scenario where conditionals are essential.

ANS: Conditionals are essential when making decisions in a program, like determining if a user input is valid.

Findings

Demonstrate understanding of the constructs through written observations.

Summary

Programming constructs provide the foundation for problem-solving. Students implemented basic constructs effectively.

Conclusion

By understanding these constructs, students are equipped to tackle programming challenges and design efficient algorithms.