Name: Brihat Ratna Bajracharya

Roll No.: 19/075

Assignment #4 (Principles of Programming Language)

Question: Explain with the examples, different types of statement level sequence controlling methods with reference to a programming language you are familiar with.

Answer

Sequence Control controls of the order of execution of the operations. The statement level sequence control determine how control flows from one part of program to another.

A. Basic Statements

Basic statements are assignments, subprogram calls, and input and output statements. Within a basic statement, sequences of operations may be invoked by using expressions.

For example:

a. assignment statement

```
a = 5
month = "february"
```

b. input/output statement

```
name = input("Enter your name: ")
print("Your name is {}".format(name))
```

c. declaration statement

```
new_list = []
new_dict = {}
```

d. break and continue statement

An early exit from a loop can be accomplished by using break statement and remaining part of the given iteration can be skipped using continue statement

B. Statement Level Sequence Control

There are three main statement-level implicit sequence controls which are:

1. Composition

In this form, statement may be placed in a textual sequence so that they are executed in order whenever the larger program structure containing the sequence is executed.

For example,

```
a = 5
b = 7
sum = a + b
print("Sum of  {0} and {1} is {2}".format(a, b, sum))
```

Above four statements are executed one after another to calculate and then print the sum of numbers in variables a and b.

2. Alternation

In this form, two sequences of statements may form alternatives so that one or the other sequence is executed, but not both.

For example,

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

if (a % 2 == 0):
    print("{} is even number.".format(a))
else:
    print("{} is odd number.".format(a))
```

Given the number a, only one part of the if-else statements ate executed depending on the condition given inside if-clause.

3. Iteration

All looping statements are iterative statements. The control repetitively executes a statement for 'n' number of times (n > 1), where 'n' is the number of iteration

For example.

```
# Example 1
for index in range(0, 10):
    print("Hello, World!")

# Example 2
fruits = ["apple", "mango", "orange", "banana"]
for fruit in fruits:
    print(fruit)
```

The first example prints the statement *"Hello, World!"* ten times in the screen while the second statement iterates over the list of fruits printing all the fruits in the list one after another.

In explicit sequence control, the default sequence is altered by some special statements like **`goto`** **`break`** and **`continue`**. There is no **`goto`** statement in Python and **`break`** and **`continue`** works in python like in C or C++.