CONTROL STRUCTURES

Control structures are fundamental components of programming languages that allow developers to control the flow of execution within their programs. These structures dictate the order in which instructions are executed based on certain conditions or criteria. By employing control structures, programmers can create complex algorithms, make decisions, and execute repetitive tasks efficiently.

Conditional Statements

Conditional statements are programming constructs that allow the execution of different blocks of code based on certain conditions. These conditions are expressions that evaluate to either true or false. Conditional statements provide the ability to make decisions within a program, enabling it to behave differently depending on the situation.

There aretwo main types of conditional statements:

if Statements: The if statement is the most basic conditional statement. It allows you to execute a block of code if a specified condition is true. If the condition evaluates to false, the code block associated with the if statement is skipped.

```
x = 10
if x > 5:
    print("x is greater than 5")
```

if-else Statements: The if-else statement allows you to execute one block of code if a condition is true, and another block of code if the condition is false.

```
x = 3
if x % 2 == 0:
    print("x is even")
else:
    print("x is odd")
```

elif Statements: The elif (short for "else if") statement allows you to check multiple conditions after an initial if statement. If the condition associated with elif evaluates to true, the corresponding block of code is executed. If none of the if or elif conditions are true, the else block (if present) is executed.

```
score = 85
if score >= 90:
```

```
    print("Grade: A")
elif score >= 80:
    print("Grade: B")
elif score >= 70:
    print("Grade: C")
else:
    print("Grade: D")
```

Switch Statement: The switch statement allows programmers to compare the value of a variable against multiple cases and execute the corresponding block of code.

```
switch (expression) {
    case value1:
        // Execute this block if expression equals value1
        break;
    case value2:
        // Execute this block if expression equals value2
        break;
    default:
        // Execute this block if expression doesn't match any case
}
```

Loops

Loops are control structures that enable repetitive execution of a block of code. They iterate over a sequence of values or until a specific condition is met. The commonly used loop structures include:

for Loop: The for loop iterates over a sequence (e.g, numbers in a range) and executes a block of code for each iteration.

while Loop: The while loop repeatedly executes a block of code as long as a specified condition is true.

do-while Loop: Unlike the while loop, the do-while loop executes the block of code first and then checks the condition. It ensures that the block of code is executed at least once.

Branching Structures

Branching structures alter the flow of execution by transferring control to a different part of the program. They include:

break Statement: The break statement is used to terminate the execution of a loop or switch statement and transfer control to the statement following the terminated loop or switch.

continue Statement: The continue statement is used to skip the current iteration of a loop and continue with the next iteration.

return Statement: The return statement is used to exit a function and return a value to the caller.

goto Statement: Although not commonly used due to its potential for creating spaghetti code, the goto statement transfers control to a labeled statement within the same function.