

COS121 Notes on:

Basic Concepts of Programming: Algorithm and Flowchart

An algorithm is a step-by-step procedure for solving a problem or achieving a specific goal. It is a set of instructions that is finite, definite, and effective. Algorithms can be expressed in various forms, such as natural language, flowcharts, pseudocode, or programming languages.

Characteristics of an Algorithm:

- Input: An algorithm receives input data.
- Processing: An algorithm performs operations on the input data.
- Output: An algorithm produces output data.
- Finiteness: An algorithm has a finite number of steps.
- Definiteness: An algorithm has well-defined steps.
- Effectiveness: An algorithm produces the desired output.

A flowchart is a graphical representation of an algorithm. It uses symbols, arrows, and text to illustrate the steps and logic of a program. Flowcharts are useful for visualizing, designing, and communicating the flow of a program.

Types of Flowchart Symbols:

- Oval: Start/Stop
- Rectangle: Process
- Diamond: Decision
- Arrow: Flow of control
- Parallelogram: Input/Output

Advantages of Flowcharts:

- Helps to break down complex problems into manageable parts.
- Ensures clarity and precision in programming.
- Facilitates communication and collaboration among programmers.
- Enhances problem-solving skills and logical thinking.

Disadvantages of Flowcharts:

- Can be time-consuming to create.
- May not be suitable for large and complex programs.
- Can be difficult to modify.

COS121 Notes on:

Keywords, Variables, and Assignment Statements

Keywords are reserved words in a programming language that have special meanings. They cannot be used as variable names. Examples of keywords include if, else, while, for, int, and double. Keywords are used to define the structure and syntax of a program.

Variables are names given to memory locations to store data. A variable has a data type, which determines the type of data that can be stored in it. Variables can be reassigned a new value. There are several types of variables, including integer variables, floating-point variables, and character variables.

Types of Variables:

Integer variables are used to store whole numbers. They are declared using the int keyword. Example: `int x = 5;`

Floating-point variables are used to store decimal numbers. They are declared using the double keyword. Example: `double y = 3.14;`

Character variables are used to store single characters. They are declared using the char keyword. Example: `char z = 'A';`

Assignment Statements:

Assignment statements are used to assign a value to a variable. The syntax for an assignment statement is `variable_name = value;`. For example: `x = 5;` assigns the value 5 to the variable x.

Rules for Naming Variables:

Variable names must start with a letter or underscore. They can contain letters, digits, and underscores. Variable names cannot contain special characters or keywords. Variable names must be unique within the program.

Examples of valid variable names include x, y, z, and totalMarks. Examples of invalid variable names include 123, if, and total-marks.

_COS121 Notes_on:

Control Structures

Control structures are used to control the flow of a program's execution. They determine which statements are executed, how many times they are executed, and in what order they are executed.

Types of Control Structures:

1. Sequential Control Structure: This is the simplest control structure, where statements are executed one after the other.

1. Conditional Control Structure: This control structure allows a program to make decisions based on certain conditions. It includes if-else statements and switch statements.

2. Repetitive Control Structure: This control structure allows a program to repeat a set of statements. It includes for loops, while loops, and do-while loops.

Conditional Control Structure:

- If-else statement: Used to execute a block of code if a condition is true, and another block of code if the condition is false.

- Switch statement: Used to execute a block of code based on the value of a variable.

Repetitive Control Structure:

- For loop: Used to repeat a block of code for a specified number of times.

- While loop: Used to repeat a block of code while a condition is true.

- Do-while loop: Used to repeat a block of code while a condition is true, but the code is executed at least once.

Example of a Conditional Control Structure:

```
if (x > 5) {  
    printf("x is greater than 5");  
} else {  
    printf("x is less than or equal to 5");  
}
```

Example of a Repetitive Control Structure:

```
for (int i = 0; i < 5; i++) {  
    printf("Hello, World!");  
}
```

[12/25, 02:09] Meta AI: _COS121 Notes_

Introduction to Arrays

An array is a collection of elements of the same data type stored in contiguous memory locations. Arrays are used to store and manipulate large amounts of data.

Characteristics of Arrays:

- All elements in an array are of the same data type.
- Each element in an array is identified by an index or subscript.
- Arrays are stored in contiguous memory locations.
- Arrays have a fixed size that is specified when the array is created.

Types of Arrays:

- One-dimensional array: A list of elements of the same data type.
- Multi-dimensional array: A table or matrix of elements of the same data type.

Declaring Arrays:

- The syntax for declaring a one-dimensional array is: `data_type array_name[array_size];`
- The syntax for declaring a multi-dimensional array is: `data_type array_name[array_size1][array_size2];`

Initializing Arrays:

- Arrays can be initialized when they are declared using the syntax: `data_type array_name[array_size] = {element1, element2, ...};`
- Arrays can also be initialized using a loop.

Accessing Array Elements:

- Array elements can be accessed using their index or subscript.
- The syntax for accessing an array element is: `array_name[index].`

Example of a One-Dimensional Array:

```
int scores[5] = {90, 80, 70, 60, 50};  
printf("%d", scores[0]); // prints 90
```

Example of a Multi-Dimensional Array:

```
int matrix[2][3] = {{1, 2, 3}, {4, 5, 6}};  
printf("%d", matrix[0][1]); // prints 2
```

_COS121 Notes_on:

Introduction to Functions

A function is a block of code that performs a specific task and can be called multiple times from different parts of a program. Functions are used to organize code, reduce repetition, and improve modularity.

Characteristics of Functions:

- A function has a unique name that identifies it.
- A function takes arguments or parameters that are passed to it when it is called.
- A function returns a value to the calling program.
- A function can be called multiple times from different parts of a program.

Types of Functions:

- Library functions: Pre-defined functions that are part of the programming language's standard library.
- User-defined functions: Functions that are defined by the programmer.

Function Declaration:

- The syntax for declaring a function is: `return_type function_name(parameter_list) { function_body }`
- The `return_type` specifies the type of value the function returns.
- The `parameter_list` specifies the names and types of the arguments the function takes.

Function Call:

- The syntax for calling a function is: `function_name(argument_list);`
- The `argument_list` specifies the values to be passed to the function.

Advantages of Functions:

- Code reusability: Functions can be called multiple times from different parts of a program.
- Code modularity: Functions break down a large program into smaller, manageable modules.
- Easier maintenance: Functions make it easier to modify or update code.

Example of a Simple Function:

```
int add(int x, int y) {  
    return x + y;  
}
```

```
int main() {  
    int result = add(5, 3);  
    printf("%d", result); // prints 8  
}
```

}

COURSE OUTLINE AND QUIZZ

COURSE OUTLINE

COS121 Course Outline

Introduction to Computer programming

Basic concept of programming

Algorithm and flowchart

Keywords ,variables and assignments

Control structures

Introduction to arrays

Introduction to functions

QUIZZ

Question 1

What is the primary goal of algorithm design in programming?

- A) To write efficient code
- B) To solve complex problems
- C) To design a user interface
- D) To optimize memory usage

Answer: B)

Question 2

What is a flowchart used for in programming?

- A) To write code
- B) To design algorithms
- C) To debug programs
- D) To test software

Answer: B)

Question 3

What is the purpose of keywords in programming?

- A) To declare variables
- B) To define functions
- C) To control program flow
- D) To identify reserved words

Answer: D)

Question 4

What is the difference between assignment and equality operators?

- A) Assignment assigns a value, while equality checks for equality
- B) Assignment checks for equality, while equality assigns a value
- C) Assignment and equality are the same
- D) Assignment is used for arithmetic operations

Answer: A)

Question 5

What type of control structure is used to execute a block of code repeatedly?

- A) Conditional statement
- B) Loop
- C) Function
- D) Array

Answer: B)

Question 6

What is the purpose of an array in programming?

- A) To store a single value
- B) To store a collection of values
- C) To perform arithmetic operations
- D) To control program flow

Answer: B)

Question 7

What is a function in programming?

- A) A block of code that performs a specific task
- B) A variable that stores a value
- C) A control structure that executes code repeatedly
- D) A data structure that stores a collection of values

Answer: A)

Question 8

What is the benefit of using functions in programming?

- A) To reduce code readability
- B) To increase code complexity
- C) To promote code reusability
- D) To decrease code efficiency

Answer: C) To promote code reusability

Question 9

What is the difference between a local variable and a global variable?

- A) Local variables are accessible globally, while global variables are accessible locally
- B) Local variables are accessible only within a function, while global variables are accessible throughout the program
- C) Local variables are used for arithmetic operations, while global variables are used for storing strings
- D) Local variables are used for storing arrays, while global variables are used for storing single values

Answer: B

Question 10

What is the purpose of comments in programming?

- A) To execute code
- B) To declare variables
- C) To explain code functionality
- D) To control program flow

Answer: C)