Section-4

Control Statements

Eng. Abdirisak Abdalla
Department of computer science

JacfuuTechno

Control Statements

- Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear.
- However, Java provides statements that can be used to control the flow of Java code.
- Such statements are called control flow statements.
- It is one of the fundamental features of Java, which provides a smooth flow of program.

Types of control flow statements.

- Java provides three types of control flow statements.
  - Decision Making statements
  - Loop statements
  - Jump statements

Decision Making statements

- As the name suggests, decision-making statements decide which statement to execute and when.
- Decision-making statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided.
- There are two types of decision-making statements in Java, i.e.,
- If statement and
- switch statement.

If Statement:

- In Java, the "if" statement is used to evaluate a condition.
- The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false.
- In Java, there are four types of if-statements given below.

➢ Simple if statement
➢ if-else statement
➢ if-else-if ladder
➢ Nested if-statement

Simple if statement

- It is the most basic statement among all control flow statements in Java.
- It evaluates a Boolean expression and enables the program to enter a block of code if the expression evaluates to true.
- Syntax of if statement is given below.
- if(condition) {
- statement 1; //executes when condition is true
- }

JacfuuTechno

if-else statement

- The if-else statement is an extension to the if-statement, which uses another block of code, i.e., else block.
- The else block is executed if the condition of the if-block is evaluated as false.
- **Syntax:**
- if(condition) {
- statement 1; //executes when condition is true
- }
- else{
- statement 2; //executes when condition is false
- }

JacfuuTechno

if-else-if ladder:

- The if-else-if statement contains the if-statement followed by multiple else-if statements.
- In other words, we can say that it is the chain of if-else statements that create a decision tree where the program may enter in the block of code where the condition is true.
- We can also define an else statement at the end of the chain.

Syntax of if-else-if statement

- if(condition 1) {
- statement 1; //executes when condition 1 is true
- }
- else if(condition 2) {
- statement 2; //executes when condition 2 is true
- }
- else {
- statement 2; //executes when all the conditions are false
- }

JacfuuTechno

Nested if-statement

- In nested if-statements, the if statement can contain a **if** or **if-else** statement inside another if or else-if statement.
- Or
- If within another if statement is called Nasted If.
- Syntax:
- if(condition 1) { statement 1;
- if(condition 2) {
- statement 2;
- } else{
- statement 3; //executes when condition 2 is false
- }
- }

JacfuuTechno

Switch Statement:

- In Java, Switch statements are similar to if-else-if statements.
- The switch statement contains multiple blocks of code called cases and a single case is executed based on the variable which is being switched.
- The switch statement is easier to use instead of if-else-if statements.
- It also enhances the readability of the program.

Points to be noted about switch statement:

- The case variables can be int, short, byte, char, or enumeration.
- String type is also supported since version 7 of Java Cases cannot be duplicate.
- Default statement is executed when any of the case doesn't match the value of expression.
- It is optional.

Con..

- Break statement terminates the switch block when the condition is satisfied.
  It is optional, if not used, next case is executed.
- While using switch statements, we must notice that the case expression will be of the same type as the variable. However, it will also be a constant value.

The syntax of switch statement

```
switch (expression){
    case value1:
     statement1;
     break;
    .
    .
    case valueN:
     statementN;
     break;
    default:
     default statement;
}
```

JacfuuTechno

Any Question?

JacfuuTechno

Thank You