FINAL REPORT

Explanations

Java Switch Statements

Use the switch statement to select one of many code blocks to be executed. This is how it works: The switch expression is evaluated once. The value of the expression is compared with the values of each case. If there is a match, the associated block of code is executed.

The break Keyword

When Java reaches a break keyword, it breaks out of the switch block. This will stop the execution of more code and case testing inside the block. When a match is found, and the job is done, it's time for a break. There is no need for more testing. A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

The default Keyword

The default keyword specifies some code to run if there is no case match. Note that if the default statement is used as the last statement in a switch block, it does not need a break.

Java String equals()

The java string equals() method compares the two given strings based on the content of the string. If any character is not matched, it returns false. If all characters are matched, it returns true. The String equals() method overrides the equals() method of Object class.

Ternary Operator

Java ternary operator is the only conditional operator that takes three operands. It’s a one-liner replacement for if-then-else statement and used a lot in Java programming. We can use the ternary operator in place of if-else conditions or even switch conditions using nested ternary operators. Although it follows the same algorithm as of if-else statement, the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.

Java Conditions and If Statements

Java supports the usual logical conditions from mathematics:

- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b
- Equal to a == b
- Not Equal to: a != b

You can use these conditions to perform different actions for different decisions. Java has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

The if Statement

Use the if statement to specify a block of Java code to be executed if a condition is true. Note that if is in lowercase letters. Uppercase letters (If or IF) will generate an error.

The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Short Hand if…Else (Ternary Operator)

There is also a short-hand if else, which is known as the ternary operator because it consists of three operands. It can be used to replace multiple lines of code with a single line.

Java Truth Table

Truth tables are used to determine whether a prepositional expression is true or false for all input values, that are, logically valid. The Truth-Table below shows the functioning of the Boolean logical operators AND (&), OR(|), XOR (^) and NOT (!).

Boolean logical operators AND (&), OR (|), XOR (^) and NOT (!).

- **& -** It produces **true** if both operands are **true** otherwise produces false.
- **| -** It produces **false** if both operands are **false** otherwise produces true.
- **^ -** It produces **true** if both operands are different otherwise produces false.
- **! -** It produces **true** if both operands are **false** and produces **false** if both operands are **true**.