**C#**

# Differences Between If-Else & Switch

* The expression inside of an if statement decides whether to execute the statements inside the if block or under the else block. For switch, the expression inside switch statement decides which case to execute.
* The if-else statement checks for equality as well as for logical expression. On the other hand, switch checks only for equality.
* The if statement evaluates integer, character, pointer or floating-point type or boolean type. On the other hand, switch statement evaluates only character or an integer datatype.
* Sequence of execution is like either statement under if block will execute or statements under else block statement will execute. However, the expression in the switch statement decides which case to execute and if you do not apply a break statement after each case it will execute till the end of switch statement.
* For an if-else statement, if the expression inside of the if turn outs to be false, the statement inside of the else block will be executed. For the switch statement, if the expression inside of the switch statement turns out to be false then the default statements are executed.
* It’s known to be difficult to edit if-else statements since it’s tedious to trace where the correction is required. Many people agree that it’s much simpler to edit switch statements since they’re easy to trace.

### Difference between Function Overloading and Function Overriding

|  |  |  |
| --- | --- | --- |
| **Basis** | **Function Overloading** | **Function Overriding** |
| Inheritance | Can happen without inheritance | Happens only when a class inherits from another class |
| Signature | Function signatures of overloaded functions must be different | Same function signatures |
| Polymorphism | Compile time | Runtime |
| Scope | Overloaded functions are in the same scope | Functions are in different scope |
| Purpose | To have multiple functions with same name that act differently depending on parameters | To perform additional or different tasks than the base class function |
| Number of times | A function can be overloaded multiple times | A function is overridden single time in its derived class |

|  |  |  |
| --- | --- | --- |
| **No.** | **StringBuffer** | **StringBuilder** |
| 1) | StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. |
| 2) | StringBuffer is *less efficient* than StringBuilder. | StringBuilder is *more efficient* than StringBuffer. |
| 3) | StringBuffer was introduced in Java 1.0 | StringBuilder was introduced in Java 1.5 |
|  |  |  |
| Basis | Array | ArrayList |
| Definition | An array is a dynamically-created object. It serves as a container that holds the constant number of values of the same type. It has a contiguous memory location. | The ArrayList is a class of Java Collections framework. It contains popular classes like Vector, HashTable, and HashMap. |
| Static/ Dynamic | Array is static in size. | ArrayList is dynamic in size. |
| Resizable | An array is a fixed-length data structure. | ArrayList is a variable-length data structure. It can be resized itself when needed. |
| Initialization | It is mandatory to provide the size of an array while initializing it directly or indirectly. | We can create an instance of ArrayList without specifying its size. Java creates ArrayList of default size. |
| Performance | It performs fast in comparison to ArrayList because of fixed size. | ArrayList is internally backed by the array in Java. The resize operation in ArrayList slows down the performance. |
| Primitive/ Generic type | An array can store both objects and primitives type. | We cannot store primitive type in ArrayList. It automatically converts primitive type to object. |
| Iterating Values | We use for loop or for each loop to iterate over an array. | We use an iterator to iterate over ArrayList. |
| Type-Safety | We cannot use generics along with array because it is not a convertible type of array. | ArrayList allows us to store only generic/ type, that's why it is type-safe. |
| Length | Array provides a length variable which denotes the length of an array. | ArrayList provides the size() method to determine the size of ArrayList. |

## Collection Hierarchy in Java

The hierarchy of the entire collection framework consists of four core interfaces such as Collection, List, Set, Map, and two specialized interfaces named SortedSet and SortedMap for sorting.

All the interfaces and classes for the collection framework are located in [java.util package](https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html). The diagram of Java collection hierarchy is shown in the below figure.

# Java Collections Framework

In this tutorial, we will learn about different interfaces of the Java collections framework.

The Java **collections** framework provides a set of interfaces and classes to implement various data structures and algorithms.

For example, the LinkedList class of the collections framework provides the implementation of the doubly-linked list data structure.

The core collection interfaces are:

Collection : The root of the collection hierarchy. ...

Set : A collection that cannot contain duplicate elements. ...

List : An ordered collection (sometimes called a sequence). ...

Queue : A collection used to hold multiple elements prior to processing. ...

Map : An object that maps keys to values.

**SQL**

1 What is SQL

* SQL stands for Structured Query Language
* SQL Lets you Access and manipulate databases
* SQL became a standard of the American Notional Standard Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

1. Categories of commands with description

Structured Query Language (SQL) as well all Know is the database language by the use of which we can perform certain operation on the existing database and also, we can use this language to create a database.  [SQL](https://www.geeksforgeeks.org/structured-query-language/) uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These [SQL](https://www.geeksforgeeks.org/sql-concepts-and-queries/)commands are mainly categorized into four categories as:

* DDL – Data Definition Language
* DQL – Data Query Language
* DML – Data Manipulation Language
* DCL – Data Control Language
* TCL \_ Transaction Control Language

1. Which Model it used

The **model** database is used as the template for all databases created on an instance of SQL Server. Because **tempdb** is created every time SQL Server is started, the **model** database must always exist on a SQL Server system.

1. Data Type in SQL

Data types are used to represent the nature of the data that can be stored in the database table. For example, in a particular column of a table, if we want to store a string type of data then we will have to declare a string data type of this column.

Data types mainly classified into three categories for every database.

* String Data types
* Numeric Data types
* Date and time Data types

**DDL Commands:**  
In this section, We will cover the following DDL commands as follows.

1. Create
2. Alter
3. truncate
4. drop

Let’s discuss it one by one.

**Command-1 :**  
**CREATE :**  
This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

**Syntax –**

CREATE TABLE table\_name

(

column\_1 datatype,

column\_2 datatype,

column\_3 datatype,

....

);

**Example –**

CREATE TABLE Student\_info

(

College\_Id number(2),

College\_name varchar(30),

Branch varchar(10)

);

**Command-2 :**  
**ALTER :**  
This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

**Syntax –**.

ALTER TABLE table\_name

ADD column\_name datatype;

**Example –**

ALTER TABLE Student\_info

ADD CGPA number;

**Command-3 :**  
**TRUNCATE :**

**Syntax –**.

TRUNCATE TABLE table\_name;

**Example –**.

TRUNCATE TABLE Student\_info;

**Command-4 :**  
**DROP :**  
This command is used to remove an existing table along with its structure from the Database.

**Syntax –**.

DROP TABLE table\_name;

**Example –**.

DROP TABLE Student\_info;

# DML Commands in SQL

DML is an abbreviation of **Data Manipulation Language**.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

**Following are the four main DML commands in SQL:**

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

## SELECT DML Command

SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

36.4M

776

Exception Handling in Java - Javatpoint

**Next**

**Stay**

**Syntax of SELECT DML command**

1. **SELECT** column\_Name\_1, column\_Name\_2, ….., column\_Name\_N **FROM** Name\_of\_table;

Here, **column\_Name\_1, column\_Name\_2, ….., column\_Name\_N** are the names of those columns whose data we want to retrieve from the table.

If we want to retrieve the data from all the columns of the table, we have to use the following SELECT command:

1. **SELECT** \* **FROM** table\_name;

### Examples of SELECT Command

**Example 1: This example shows all the values of every column from the table.**

1. **SELECT** \* **FROM** Student;

**Example 2: This example shows all the values of a specific column from the table.**

**1. SELECT** Emp\_Id, Emp\_Salary **FROM** Employee;

**Example 3: This example describes how to use the WHERE clause with the SELECT DML command.**

If you want to access all the records of those students whose marks is 80 from the above table, then you have to write the following DML command in SQL:

1. **SELECT** \* **FROM** Student **WHERE** Stu\_Marks = 80;

## INSERT DML Command

INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.

**Syntax of INSERT Command**

1. **INSERT** **INTO** TABLE\_NAME ( column\_Name1 , column\_Name2 , column\_Name3 , .... column\_NameN )  **VALUES** (value\_1, value\_2, value\_3, .... value\_N ) ;
2. **INSERT** **INTO** Student (Stu\_id, Stu\_Name, Stu\_Marks, Stu\_Age) **VALUES** (104, Anmol, 89, 19);

## UPDATE DML Command

UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

**Syntax of UPDATE Command**

1. **UPDATE** Table\_name **SET** [column\_name1= value\_1, ….., column\_nameN = value\_N] **WHERE** CONDITION;

### Here, 'UPDATE', 'SET', and 'WHERE' are the SQL keywords, and 'Table\_name' is

the name of the table whose values you want to update.

1. **UPDATE** Product **SET** Product\_Price = 80 **WHERE** Product\_Id = 'P102' ;
2. **UPDATE** Student **SET** Stu\_Marks = 80, Stu\_Age = 21 **WHERE** Stu\_Id = 103 AND Stu\_Id = 202;

## DELETE DML Command

DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

**Syntax of DELETE Command**

1. **DELETE** **FROM** Table\_Name **WHERE** condition;

### Examples of DELETE Command

**Example 1: This example describes how to delete a single record from the table.**

Let's take a Product table consisting of the following records:

|  |  |  |  |
| --- | --- | --- | --- |
| **Product\_Id** | **Product\_Name** | **Product\_Price** | **Product\_Quantity** |
| P101 | Chips | 20 | 20 |
| P102 | Chocolates | 60 | 40 |
| P103 | Maggi | 75 | 5 |
| P201 | Biscuits | 80 | 20 |
| P203 | Namkeen | 40 | 50 |

Suppose, you want to delete that product from the Product table whose Product\_Id is P203. To do this, you have to write the following DML DELETE command:

1. **DELETE** **FROM** Product **WHERE** Product\_Id = 'P202' ;

**Example 2: This example describes how to delete the multiple records or rows from the database table.**

Let's take a Student table consisting of the following records:

|  |  |  |  |
| --- | --- | --- | --- |
| **Stu\_Id** | **Stu\_Name** | **Stu\_Marks** | **Stu\_Age** |
| 101 | Ramesh | 92 | 20 |
| 201 | Jatin | 83 | 19 |
| 202 | Anuj | 85 | 19 |
| 203 | Monty | 95 | 21 |
| 102 | Saket | 65 | 21 |
| 103 | Sumit | 78 | 19 |
| 104 | Ashish | 98 | 20 |

Suppose, you want to delete the record of those students whose Marks is greater than 70. To do this, you have to write the following DML Update command:

1. **DELETE** **FROM** Student **WHERE** Stu\_Marks > 70 ;

Data control language (DCL) is used to access the stored data. It is mainly used for revoke and to grant the user the required access to a database. In the database, this language does not have the feature of rollback.

* It is a part of the structured query language (SQL).
* It helps in controlling access to information stored in a database. It complements the data manipulation language (DML) and the data definition language (DDL).
* It is the simplest among three commands.
* It provides the administrators, to remove and set database permissions to desired users as needed.
* These commands are employed to grant, remove and deny permissions to users for retrieving and manipulating a database.

## DDL Commands

The Data Definition Language (DDL) commands are as follows −

### GRANT Command

It is employed to grant a privilege to a user. GRANT command allows specified users to perform specified tasks

**Syntax**

GRANT privilege\_name on objectname to user;

Here,

* privilege names are SELECT,UPDATE,DELETE,INSERT,ALTER,ALL
* objectname is table name
* user is the name of the user to whom we grant privileges

### REVOKE Command

It is employed to remove a privilege from a user. REVOKE helps the owner to cancel previously granted permissions.

**Syntax**

 REVOKE privilege\_name on objectname from user;

Here,

* privilege names are SELECT,UPDATE,DELETE,INSERT,ALTER,ALL
* objectname is table name
* user is the name of the user whose privileges are removing

### Example

GRANT SELECT, UPDATE ON employees TO Bhanu

Explanation − Firstly, to give the permissions to user, we have to use GRANT command. The privileges are SELECT because to view the records and UPDATE to modify the records. The objectname is table name which is Employee. The user name is bhanu.

REVOKE SELECT, UPDATE ON employees TO Bhanu

Explanation − Firstly, to revoke the permissions to user, we have to use REVOKE command. The privileges Need to revoke are SELECT because to view the records and UPDATE to modify the records. The objectname is table name which is Employee. The user name is Bhanu.

1. Truncate vs drop vs delete