

.NET > C#.NET > OOP

Delegates

- by Harsha Vardhan

Delegates

- "Delegate" is a "type" that represents methods that have specific parameters and return type.
- The "delegate object" (a.k.a. delegate instance), is an object that stores reference (address) of a specific method of a specific class, with compatible parameters and return type, which is already defined in the delegate type.
- You can invoke the methods using delegate objects.
- Delegates are used to pass methods as arguments to other methods.

Creating Delegate

1

```
public delegate ReturnType DelegateName(param1, param2, ...);
```

Creating Delegate Object

2

```
DelegateName ReferenceVariable = MethodName;
```

Invoke Method using Delegate Object

3

```
ReferenceVariable( );
```

Types of Delegates

- **Single Cast Delegates**
 - Contains reference of only one method.
 - When called, it directly invokes the referenced method.
- **Multi Cast Delegates**
 - Contains references of multiple methods.
 - When called, it invokes all the referenced methods, one-by-one in a sequence.

Rules for Delegates

- The method signature (parameters and return type) must match between the "method" and "delegate".
- Delegates can be used as "parameter type" or "return type" of a method.
- You can store references of non-static method or static method in the delegate object.
- The methods, which reference is stored in the "single-cast delegate object", can have return value.

- The methods, which reference is stored in the "multi-cast delegate object", can't have return value; in case, if they have return value, the return value of lastly-executed method only can be received; others will be ignored.