

Delegates

What

- › "Delegate type" is a "type" that represents methods that have specific parameters and return type.
- › The "delegate" (a.k.a. delegate object), 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.

How

Creating Delegate Type

1

```
public delegate Return Type DelegateTypeName(param1, param2, ...);
```

Creating Delegate Object

2

```
DelegateTypeName ReferenceVariable = new DelegateTypeName (MethodName);
```

Invoke Method using Delegate Object

3

```
ReferenceVariable.Invoke( arg1, arg2, ... );
```

Rules

- › You can invoke the methods using 'delegate objects' (or) 'delegates'.
- › Delegates are used to pass methods as arguments to other methods.
- › 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.
- › All delegate types are derived from "System.Delegate" class.

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.
- › All methods' parameters and return type should be same.