

Task 7: Exception Handling

(Scope: User Defined Exception/Checked /Unchecked Exception/Exception handling using try..catchfinally,throw & throws keyword usage)

Define the following custom exceptions and throw them in methods whenever needed . Handle all the exceptions in main method,

1. **TrackingNumberNotFoundException** :throw this exception when user try to withdraw amount or transfer amount to another acco
2. **InvalidEmployeeIdException** throw this exception when id entered for the employee not existing in the system

Code:

Exceptions.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CMS.Exceptions
{

    public class TrackingNumberNotFoundException : Exception
    {
        public TrackingNumberNotFoundException(string message) : base(message) { }
    }

    public class InvalidEmployeeIdException : Exception
    {
        public InvalidEmployeeIdException(string message) : base(message) { }
    }
}
```

CourierUserService.cs (modified fuctions)

```
public string GetOrderStatus(string trackingNumber)
{
    var order = courierOrders.FirstOrDefault(c => c.TrackingNumber == trackingNumber);

    if (order == null)
        throw new TrackingNumberNotFoundException("Order with the given tracking number not found.");

    return order.Status;
}

public bool CancelOrder(string trackingNumber)
{
    var order = courierOrders.FirstOrDefault(c => c.TrackingNumber == trackingNumber);
    if (order == null)
        throw new TrackingNumberNotFoundException("Tracking number not found.");

    if (order.Status != "Delivered")
    {
        order.Status = "Cancelled";
        return true;
    }
    return false;
}
```