1. Create Demo Application to replicate memory issue
   1. Create new Windows Form Application
   2. Create method to open connection & execute command “Select getdate()”
   3. Add button with text “Execute” on it
   4. Add Execute button click event handler to call above method 1000 times to see SQL connection pool error
2. Implement IDisposable interface
   1. Open application created in previous assignment
   2. Implement IDisposable interface
   3. Add Dispose method to add console message to display connectionstrnig object hash code during disposal
   4. Observe object disposal event execution. Note that dispose message logging hashcode is not getting executed as objects are not being disposed immediately
3. Setting MaxPoolSize & ConnectionTimeOut properties of connection
   1. Add MaxPoolSize = 300 & ConnectionTimeOut =2 properties to connection
   2. Run the application again to see there is no connection pool error
4. Implement Using statement to dispose connection object
   1. Open application created in previous assignment
   2. Wrap connection with using statement
   3. Run the application executable multiple times to see that SQL connection pool error is no more
5. Using DotMemory Profiler
   1. Modify application created in Exercise 1 (i.w. without IDisposable interface) to read connection string from config file
   2. Make connectonstring object to be static field toinitialize if null
   3. Open DotMemory profiler
   4. Start Application through profiler
   5. Click Execute button to run getdate query
   6. Take profiler snapshot
   7. Click Execute button for 10 more times
   8. Take profiler snapshot again
   9. Compare number of objects being created &memory usage in both snapshots
   10. Modify application to use local variable instead of static field for connectionstring object
   11. Repeat steps e to I
6. Using DotMemory Profiler with IDisposable interface without finalization
   1. Open application used in previous exercise
   2. Wrap connectionstrng object with using statemet
   3. Implement IDisposable interface
   4. Override Dispose method with Boolean parameter
   5. Add Dispose method to call this overridden dispose method with parameter value ”true”. Also add statement GC.SupressFinalize(this)” to this method
   6. In the overridden Dispose method add below code –

If(disposing)

If(\_connection != null)

{

\_connection.dispose();

\_connection = null;

]

* 1. Open DotMemory profiler
  2. Start Application through profiler
  3. Click Execute button to run getdate query
  4. Take profiler snapshot
  5. Click Execute button for 10 more times
  6. Take profiler snapshot again
  7. Compare number of objects being created &memory usage in both snapshots
  8. Modify application to use local variable instead of static field for connectionstring object
  9. Repeat steps e to I