**Question 1:** Implement Singleton pattern on printer functionality.

**Code:**

**Main:**

Singleton obj = Singleton.GetObject();

obj.Print("Singleton pattern on printer functionality");

Console.WriteLine("Enter number of printer connected");

int number = int.Parse(Console.ReadLine());

obj.Printer(number);

obj.Print("Good Bye...!");

Console.ReadLine();\_

**Singleton class:**

protected static Singleton \_obj;

private Singleton() { }\_

public static Singleton GetObject()

{

if (\_obj == null)

{

\_obj = new Singleton();

}

return \_obj;

}

public void Printer(int num)

{

Console.WriteLine("{0} Printer are connected:",num);

Console.WriteLine("Enter Name of printers!");

string[] name = new string[num];

for (int i = 0; i < name.Length; i++)

{

name[i] = Console.ReadLine();

}

Console.WriteLine("Printer in Queue");

for (int j = 0; j < name.Length; j++)

{

Console.WriteLine("Printer No {0} = {1}",j+1,name[j]);

}

}

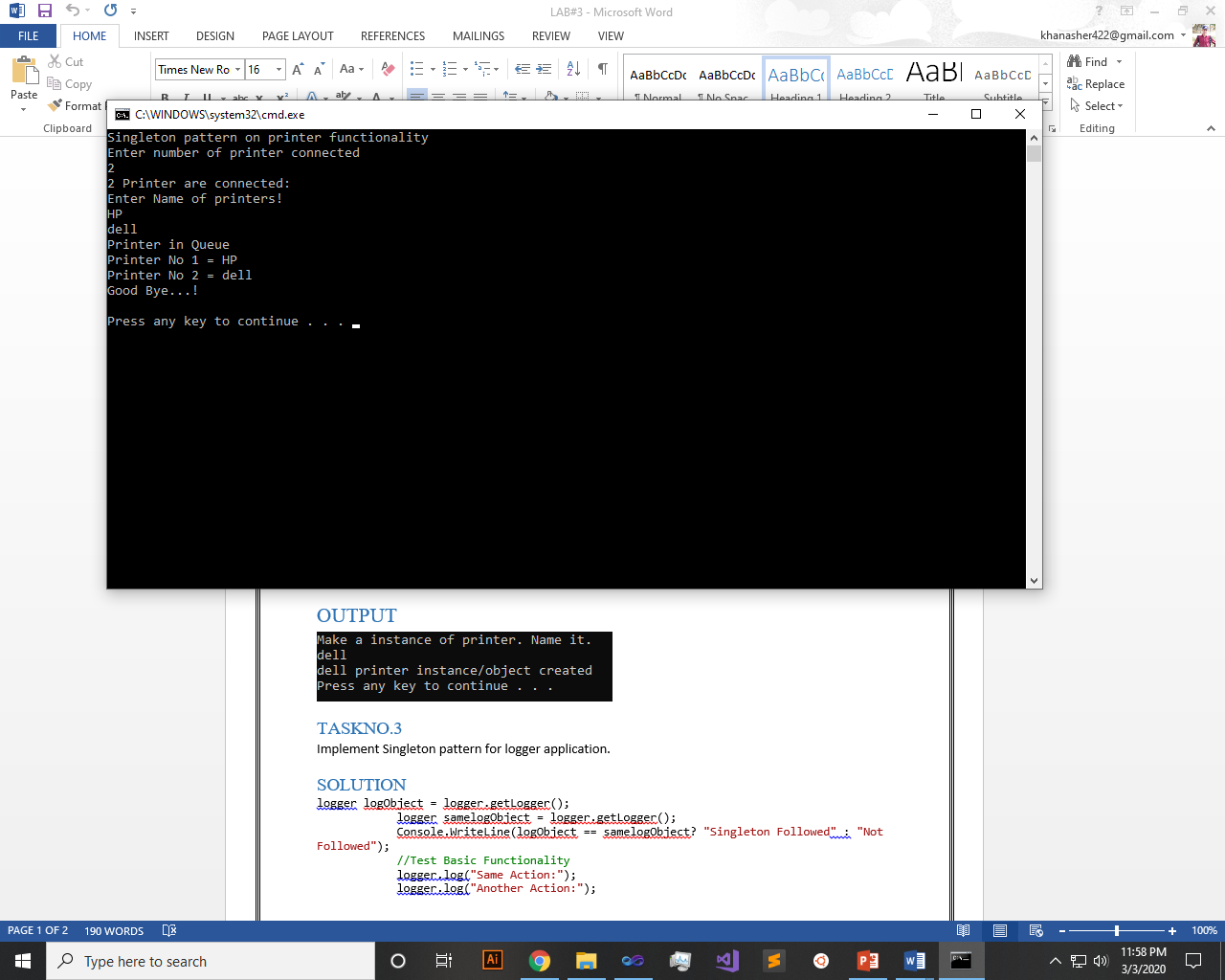
public void Print(string s)

{

Console.WriteLine(s);

}

**Output:**



**Question 2:** Implement Singleton pattern for Sessions.

**Code:**

**Main:**

Session obj = Session.GetObject();

obj.Print("Singleton pattern for Session");

Console.WriteLine("Enter number of Sessions");

int number = int.Parse(Console.ReadLine());

obj.Printer(number);

obj.Print("Good Bye...!");

Console.ReadLine();

**Session class:**

protected static Session \_obj;

private Session() { }\_

public static Session GetObject()

{

if (\_obj == null)

{

\_obj = new Session();

}

return \_obj;

}

public void Printer(int num)

{

Console.WriteLine("{0} Sessions are connected or exist:",num);

Console.WriteLine("Enter Name of Sessions!");

string[] name = new string[num];

for (int i = 0; i < name.Length; i++)

{

name[i] = Console.ReadLine();

}

Console.WriteLine("Each Sessions in a Sequence");

for (int j = 0; j < name.Length; j++)

{

Console.WriteLine("Session {0} = {1}",j,name[j]);

}

}

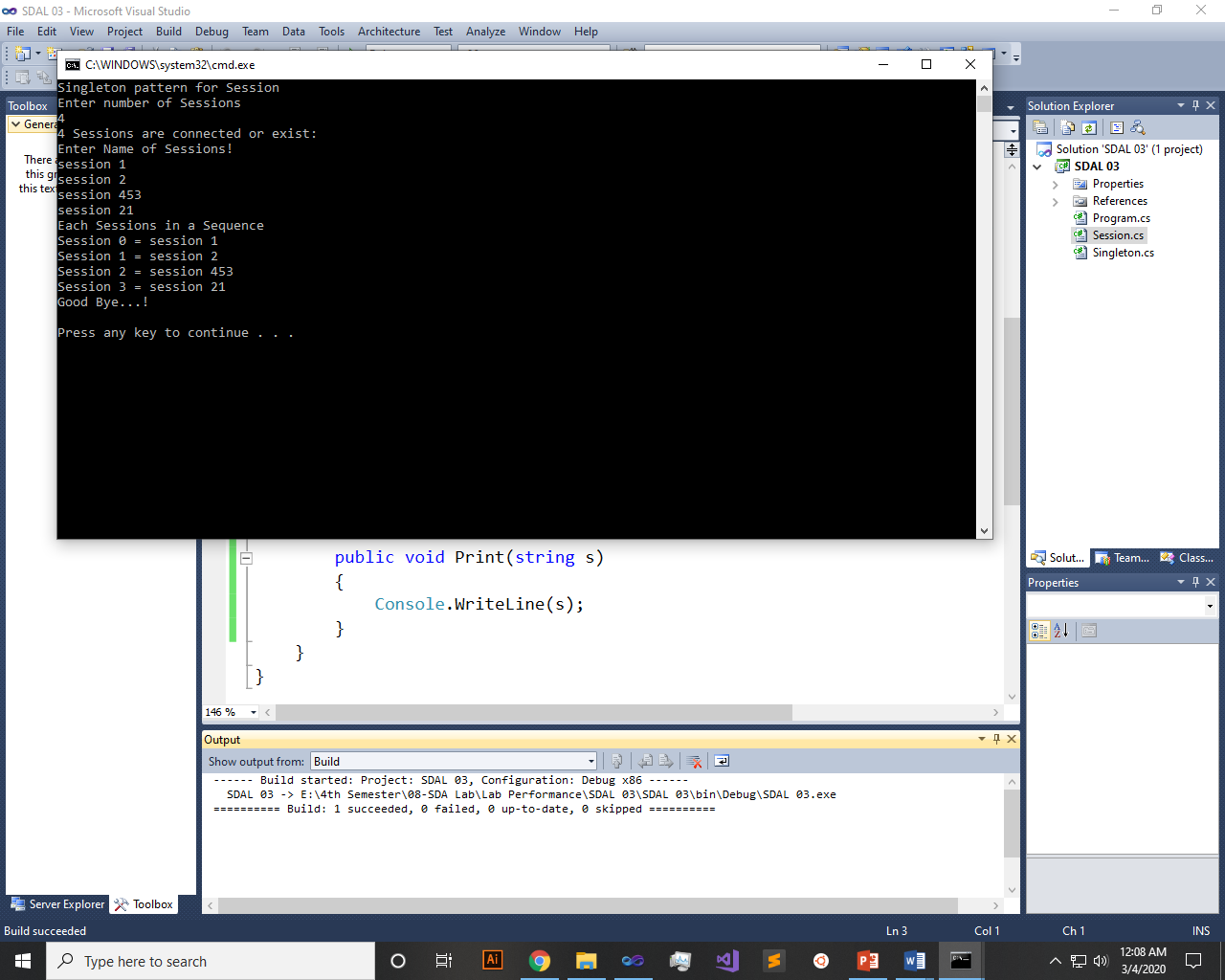
public void Print(string s)

{

Console.WriteLine(s);

}

**Output:**



**Question 3:** Implement Singleton pattern for logger application.

**Code:**

**Main:**

Logger obj = Logger.GetObject();

obj.Print("Singleton pattern for Logger");

Console.WriteLine("Enter number of Logs");

int number = int.Parse(Console.ReadLine());

obj.Printer(number);

obj.Print("Good Bye...!");

Console.ReadLine();

**Logger Class:**

protected static Logger \_obj;

private Logger() { }

public static Logger GetObject()

{

if (\_obj == null)

{

\_obj = new Logger();

}

return \_obj;

}

public void Printer(int num)

{

Console.WriteLine("{0} Logger app are connected or exist:",num);

Console.WriteLine("Enter Name of logs!");

string[] name = new string[num];

for (int i = 0; i < name.Length; i++)

{

name[i] = Console.ReadLine();

}

Console.WriteLine("Each logs in a Sequence");

for (int j = 0; j < name.Length; j++)

{

Console.WriteLine("Date n Time : {2} : Log {0} = {1}",j,name[j],DateTime.Now);

}

}

public void Print(string s)

{

Console.WriteLine(s);

}

**Output:**

