KANDULA SRINIVASA REDDY MEMORIAL COLLEGE OF ENGINEERING
(AUTONOMOUS)

KADAPA-516003. AP

(Approved by AICTE, Affiliated to JNTUA, Ananthapuramu, Accredited by NAAC)

(An ISO 9001-2008 Certified Institution)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATION COURSE
ON
".NET TECHNOLOGY"

Resource Person : 1. Mrs. B. Gouri, Assistant Professor, Dept. of CSE, KSRMCE

2. Mr. Y. Prasad Reddy, Assistant Professor, Dept. of CSE, KSRMCE

Course Coordinator: Mr. S. Khaja Khizar , Assistant Professor, Dept. of CSE, KSRMCE

Duration: 19/07/2019 to 06/08/2019

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Lr./KSRMCE/ (Department of CSE)/2019-20

Date: 15/07/2019

To
The Principal
KSRM College of Engineering
Kadapa, AP.

Sub: KSRMCE - (Department of CSE) – Permission to conduct certification course on .Net Technology - Requested – reg.

---***---

Respected Sir,

With reference to the cited, the Department of CSE is planning to conduct certificate course on **".Net Technology"** for B.Tech students from 19/07/2019 to 06/08/2019. So I request you to grant permission to conduct the certificate course. This is submitted for your kind perusal.

Thanking you sir,

Yours Faithfully,

S. Khaja Khizar

Coordinator,
S. Khaja Khizar
Assistant Professor,
CSE Dept.,

Forwarded to the Principal Sir,

Cc:

To The Director for Information

To All Deans/HODs

Permitted
U.S.S. Murty
15/07/2019

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Dated: 16/07/2019

Circular

All the B.Tech students are here by informed that department of Computer Science & Engineering is going to organize certification course on **".Net Technology"** from 19/07/2019 to 06/08/2019. Interested students do register their names with the below mentioned coordinator on or before 17/07/2019, 5 PM.

For any queries contact,

Coordinator
Sri. S. Khaja Khizar ,
Assistant Professor,
CSE Dept.,

HoD

Dr. M. Sreenivasulu,
M. E., Ph. D.
Professor & HOD CSE
K.S.R.M College of Engineering
KADAPA - 516 003

Cc to:

The Management /Director / All Deans / All HODS/Staff / Students for information

The IQAC Cell for Documentation

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Date:16/07/2019

Department of Computer Science & Engineering

Certificate Course on .Net Technology (19/07/2019 to 06/08/2019)

Registered Student List

S.No.	Roll Number	Name Of The Student	Year & Branch	Signature

Coordinator **HoD**

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Date:16/07/2019

Department of Computer Science & Engineering

Certificate Course on .Net Technology (19/07/2019 to 06/08/2019)

Registered Student List

S.No.	Roll Number	Name Of The Student	Year & Branch	Signature
1	179Y1A0534	D. Yogya vignesh	V Sem	Yogya
2	179Y1A0562	k. Venkata Sai kumar Reddy	V sem	Venkatasai kumar reddy
3	179Y1A0531	D. Yogesh	B Tech V Sem	D. Yogesh
4	179Y1A0560	K. Kavitha Reddy	B. Tech V Sem	K. Kavitha
5	179Y1A0548	G. Vinod Kumar	B. Tech V Sem	Gvinod
6	179Y1A0552	G. Ravindhar Reddy	V Sem.	Ravi
7	179Y1A0548	G. Haritha.	B. Tech V Sem	G. Haritha
8	169Y1A0515	C. Bagya Sree	VII Sem	Bagya.
9	169Y1A0509	B. Venkatesh	B. Tech VII Sem	B. Venkatesh
10	179Y1A0539	D. Devananda	B Tech V Sem	Deva
11	169Y1A0502	A. Sreekanth	VII Sem	Sree
12	179Y1A0558	J. Sai Kumar	V Sem	Sai
13	179Y1A0557	J. HARITHA	V Sem	J. Haritha
14	169Y1A0563	M. Ajay Kumar	VII Sem	Ajay
15	169Y1A0564	Mo. Dinesh Reddy	VII Sem	Dineshreddy

16	169Y1A0556	K. Supraja	B. Tech VII Sem	K. SUPRAJA
17	179Y1A0564	Kancharla Sasikala	B. Tech V Sem	K. Sasi
18	169Y1A0566	M. Sai Yashwanth.	VII sem	Sai
19	179Y1A0566	K. Harika.	V sem	Harika
20	169Y1A0565	M. Suresh	VII sem	Suresh
21	169Y1A0546	K. Narasimha.	VII sem	Narasimha
22	179Y1A0540	D. Yasaswini	V sem.	Yaswini.
23	179Y1A0561	K. Varshitha Reddy	V Sem	K. Varshitha
24	179Y1A0540	D. Yasawini	V Sem	Yasawini
26	179Y1A0526	C.V.S. Kowshiknath Reddy	V sem	Kowshiknath.
27	169Y1A0532	G. Upendra	VII Sem	Upendra
28	179Y1A0527	C. Brahmani	V. sem	Brahmani
29	169Y1A0559	K. NARESH	VII SEM	K. NARESH
30	179Y1A0531	C. Akhila	V sem	Akhila.
31	169Y1A0534	G. Krishna kanth	VII Sem	Krishna.
32	179Y1A0543	D. Jyothsna	V sem	jyothsna
33	179X1A0547	G. Sudeesh chandra Rao	V sem	Sudesh
34	179Y1A0532	D. Mohammmad Sharuk	V Sem	Sharuk.
35	179Y1A0553	G. Sai Sujitha Reddy	V Sem B Tech	G. Saisujitha
36	179Y1A0528	C. Shaik mahammed Aslam	V sem	Shaik.
37	169Y1A0563	M. Ajay Kumar	B. Tech VII Sem	Ajay Kuma
38	179Y1A0538	D. Anusha	V Sem	D Anusha
39	179Y1A0528	C. Mahamad rafi	V sem.	Mahamad
40	179Y1A0554	E. Abishek kumar Sahu	V Sem B. Tech	E. Abi

41	179YA0585	A. Nagamohan Reddy	V.Sem	Mohan
42	179Y1A0516	B. Susmita Reddy	V sem	B. Somitha
43	179Y1A0523	C. Harinadh Reddy	V sem	Harinadh
44	179YA0510	B. Divya Sree.	V sem	Divya
45	179Y1A0524	C. Harsha vardan Reddy	V sem	Harsha
46	179Y1A0517	B. prabhanjan Reddy	V sem	B. prabhan
47	179YA0506	A. Sai Charan Yadav	V.sem	Sai
48	179Y1A0518	B. Ganeshnaidu.	V sem	B. Ganeshnaidu
49	179YA0508	B. Siva. chandra.	V sem.	Siva
50	179Y1A0525	C. Sravani	V sem	Sravani
51	179YA0571	B. Govardhan Reddy	V sem	Govardhan
52	179YA0502	A. Srinath	V sem	Sri
53	179Y1A0519	B. Reddyprasad	V sem	Reddyprasad
54	179YA0513	B. Manhar Reddy	V sem	Manhar
55	179YA0515	B. Bhargavi	V sem	Bhargavi
56	179YA0509	B. Sai Kishor	V sem.	Sai
57	179Y1A0520	B. BHARATHI	V sem	Bharathi
58	179YA0512	B. G. Vikram Reddy	V sem.	Vikram
59	199Y1A0522	C. Lohith Kumar	V sem	Lohitkumar
60	179YA0507	A. Nagalakshmi	V sem	Naga
61	179YA0533	D. Sowmya.	V sem	Sowmya.

Coordinator

HoD

Dr. M. Sreenivasulu,
M. E., Ph. D.
Professor & HOD CSE
K.S.R.M. College of Engineering
KADAPA - 516 003

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Department of Computer Science & Engineering

Certificate Course on .net Technology

Registered Student List

S.No.	Roll Number	NAME OF THE STUDENT	Year & Branch	Email id
1	179Y1A0502	Adimulam Srinath	B.Tech V Sem	179Y1A0502@ksrmce.ac.in
2	179Y1A0505	Ammireddy Nagamohan Reddy	B.Tech V Sem	179Y1A0505@ksrmce.ac.in
3	179Y1A0506	Anandi Sai Charan Yadav	B.Tech V Sem	179Y1A0506@ksrmce.ac.in
4	179Y1A0507	Avula Nagalakshmi	B.Tech V Sem	179Y1A0507@ksrmce.ac.in
5	179Y1A0508	Balagani Sivachandra	B.Tech V Sem	179Y1A0508@ksrmce.ac.in
6	179Y1A0509	Balla Saikishor	B.Tech V Sem	179Y1A0509@ksrmce.ac.in
7	179Y1A0510	Bandaru Divya Sree	B.Tech V Sem	179Y1A0510@ksrmce.ac.in
8	179Y1A0511	Bandepureddy Govardhan Reddy	B.Tech V Sem	179Y1A0511@ksrmce.ac.in
9	179Y1A0512	Basireddu Gari Vikram Reddy	B.Tech V Sem	179Y1A0512@ksrmce.ac.in
10	179Y1A0513	Battina Manohar Reddy	B.Tech V Sem	179Y1A0513@ksrmce.ac.in
11	179Y1A0515	Bhumireddy Bhargavi	B.Tech V Sem	179Y1A0515@ksrmce.ac.in
12	179Y1A0516	B.Susmitha Priya	B.Tech V Sem	179Y1A0516@ksrmce.ac.in
13	179Y1A0517	Bojja Prabhanjan Reddy	B.Tech V Sem	179Y1A0517@ksrmce.ac.in
14	179Y1A0518	Bokkasam Ganeshnaidu	B.Tech V Sem	179Y1A0518@ksrmce.ac.in
15	179Y1A0519	Bonamsetty Reddyprasad	B.Tech V Sem	179Y1A0519@ksrmce.ac.in
16	179Y1A0520	Bosigari Bharathi	B.Tech V Sem	179Y1A0520@ksrmce.ac.in
17	179Y1A0522	C.Lohith Kumar	B.Tech V Sem	179Y1A0522@ksrmce.ac.in
18	179Y1A0523	Chanda Harinadh Reddy	B.Tech V Sem	179Y1A0523@ksrmce.ac.in
19	179Y1A0524	Chappidi Harsha Vardhan Reddy	B.Tech V Sem	179Y1A0524@ksrmce.ac.in
20	179Y1A0525	Chavva Sravani	B.Tech V Sem	179Y1A0525@ksrmce.ac.in
21	179Y1A0526	C.V.S.Kowshiknath Reddy	B.Tech V Sem	179Y1A0526@ksrmce.ac.in
22	179Y1A0527	Chenuru Brahmani	B.Tech V Sem	179Y1A0527@ksrmce.ac.in
23	179Y1A0528	Cheppali Shaik Mahammed Aslam	B.Tech V Sem	179Y1A0528@ksrmce.ac.in
24	179Y1A0529	Chilakala Mahammadrafi	B.Tech V Sem	179Y1A0529@ksrmce.ac.in
25	179Y1A0531	Chinthala Akhila	B.Tech V Sem	179Y1A0531@ksrmce.ac.in
26	179Y1A0532	D Mohammmad Sharuk	B.Tech V Sem	179Y1A0532@ksrmce.ac.in
27	179Y1A0533	Dabbera Sowmya	B.Tech V Sem	179Y1A0533@ksrmce.ac.in

28	179Y1A0534	Daddanala Yogya Vignesh	B.Tech V Sem	179Y1A0534@ksrmce.ac.in
29	179Y1A0537	D. Yogesh	B.Tech V Sem	179Y1A0537@ksrmce.ac.in
30	179Y1A0538	D Anusha	B.Tech V Sem	179Y1A0538@ksrmce.ac.in
31	179Y1A0539	Doddi Devananda	B.Tech V Sem	179Y1A0539@ksrmce.ac.in
32	179Y1A0540	D.Yasaswini	B.Tech V Sem	179Y1A0540@ksrmce.ac.in
33	179Y1A0543	D. Jyothsna	B.Tech V Sem	179Y1A0543@ksrmce.ac.in
34	179Y1A0544	E. Abhishek Kumar Sahu	B.Tech V Sem	179Y1A0544@ksrmce.ac.in
35	179Y1A0547	G. Sudeesh Chandra Rao	B.Tech V Sem	179Y1A0547@ksrmce.ac.in
36	179Y1A0548	G. Haritha	B.Tech V Sem	179Y1A0548@ksrmce.ac.in
37	179Y1A0549	G. Vinod Kumar	B.Tech V Sem	179Y1A0549@ksrmce.ac.in
38	179Y1A0552	G. Rajavardhan Reddy	B.Tech V Sem	179Y1A0552@ksrmce.ac.in
39	179Y1A0553	G. Sai Sujitha Reddy	B.Tech V Sem	179Y1A0553@ksrmce.ac.in
40	179Y1A0557	J. Haritha	B.Tech V Sem	179Y1A0557@ksrmce.ac.in
41	179Y1A0558	J Sai Kumar	B.Tech V Sem	179Y1A0558@ksrmce.ac.in
42	179Y1A0560	K Kavitha Reddy	B.Tech V Sem	179Y1A0560@ksrmce.ac.in
43	179Y1A0561	K Varshitha Reddy	B.Tech V Sem	179Y1A0561@ksrmce.ac.in
44	179Y1A0562	Kambham Venkata Sai Kumar Reddy	B.Tech V Sem	179Y1A0562@ksrmce.ac.in
45	179Y1A0564	Kancharla Sasikala	B.Tech V Sem	179Y1A0564@ksrmce.ac.in
46	179Y1A0566	Kanchaveerla Harika	B.Tech V Sem	179Y1A0566@ksrmce.ac.in
47	169Y1A0502	A. Sreekanth	B.Tech VII Sem	169Y1A0502@ksrmce.ac.in
48	169Y1A0509	B. Venkatesh	B.Tech VII Sem	169Y1A0509@ksrmce.ac.in
49	169Y1A0515	C. Bagya Sree	B.Tech VII Sem	169Y1A0515@ksrmce.ac.in
50	169Y1A0528	D. Jyoshna	B.Tech VII Sem	169Y1A0528@ksrmce.ac.in
51	169Y1A0532	G. Upendra	B.Tech VII Sem	169Y1A0532@ksrmce.ac.in
52	169Y1A0534	G. Krishna Kanth	B.Tech VII Sem	169Y1A0534@ksrmce.ac.in
53	169Y1A0546	K. Narasimha	B.Tech VII Sem	169Y1A0546@ksrmce.ac.in
54	169Y1A0556	K. Supraja	B.Tech VII Sem	169Y1A0556@ksrmce.ac.in
55	169Y1A0559	K. Naresh	B.Tech VII Sem	169Y1A0559@ksrmce.ac.in
56	169Y1A0562	M. Srinivasulu	B.Tech VII Sem	169Y1A0562@ksrmce.ac.in
57	169Y1A0563	M. Ajay Kumar	B.Tech VII Sem	169Y1A0563@ksrmce.ac.in
58	169Y1A0564	M. Dinesh Reddy	B.Tech VII Sem	169Y1A0564@ksrmce.ac.in
59	169Y1A0565	M. Suresh	B.Tech VII Sem	169Y1A0565@ksrmce.ac.in
60	169Y1A0566	M. Sai Yashwanth	B.Tech VII Sem	169Y1A0566@ksrmce.ac.in

Coordinator

HoD

Dr. M. Sreenivasulu,
M. E., Ph. D.
Professer & HOD CSE
K.S.R.M. College of Engineering
KADAPA - 516 003

K.S.R.M. College of Engineering (Autonomous), Kadapa.

Department of CSE

Certificate Course on

.Net Technology

Syllabus

Overview: The course is designed to provide complete knowledge on .net technology and to enhance the programming skills of the students by giving practical assignments to be done in labs.

Course Objectives: . Net is a framework that provides programming guidelines that can be used to develop a wide range of applications from web to mobile to windows-based application.

Course Outcomes: At the end of the course participants will be able to

1. Understand the basics of VB.NET&ASP.NET
2. Understand the concept of error and exception handling.
3. Understand the OOPS Concepts.

Module 1: Introduction to the .NET initiative and the .NET platform

Module 2: Code Management

Module 3: Object oriented programming concepts and Error and Exception handling

Module 4: Using controls and components for designing application.

Module 5: Design web application using ASP.net

Textbook:

- C# 6.0 and the .NET 4.6 Framework by Andrew Troelsen and Philip Japikse.
- Programming Entity Framework by Julia Lerman.
- Pro ASP.Net MVC 5 (Expert's Voice in ASP.Net) by Adam Freeman.
- Dependency Injection in .NET by Mark Seemann.
- C# in Depth by Jon Skeet.

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Department of Computer Science & Engineering

Certificate Course on .Net Technology (19/07/2019 to 06/08/2019)

Schedule

S.No	Date	Time	Faculty	Topic
1	19/07/2019	3PM to 5PM	Sri. S. Khaja Khizar & Sri. Y. Prasad Reddy	Module I: Introduction to the .net initiative and the .net platform: • Detailed .NET Framework architecture, Name spaces • Assemblies, Exploring Assemblies and Namespaces • Common Language Implementation , Metadata and Intermediate Language
		5PM to 6PM	Sri. Y. Prasad Reddy	• Garbage Collection, Versioning and Side-by-Side Execution, The End to DLL Hell, Managed Execution, COM Inter Op.
2	20/07/2019	3PM to 5PM	Smt. B. Gouri	Installation procedure for .net technology.
		5PM to 6PM	Smt. B. Gouri	Module II: Code Management Data Types and Base Class Libraries - Understanding .NET Data Types - Stream and String Manipulation,
3	22/07/2019	3PM to 5PM	Smt. B. Gouri	Files and I/O, Collections - The Microsoft. Visual Basic Namespace
		5PM to 6PM	Smt. B. Gouri	Module III: Object Oriented Programming Concepts: Introduction to VB.NET and C#, Namespaces
4	23/07/2019	3PM to 5PM	Smt. B. Gouri	Creating Classes, Classes and Inheritance
		5PM to 6PM	Smt. B. Gouri	Overloading
5	24/07/2019	4PM to 5PM	Smt. B. Gouri	Constructors and Destructors
		5PM to 6PM	Smt. B. Gouri	Inheritance
6	25/07/2019	3PM to 5PM	Smt. B. Gouri	Controlling scope and visibility, Dispose and Finalization, Modifiers, Properties and

				Indexers
		5PM to 6PM	Smt. B. Gouri	Attributes, Reflection API
7	26/07/2019	4PM to 5PM	Sri. Y. Prasad Reddy	Module IV: Language Features Creating .NET Projects, Namespaces,
		5PM to 6PM	Sri. Y. Prasad Reddy	Threads
8	27/07/2019	4PM to 5PM	Sri. Y. Prasad Reddy	Debugging and Error Handling,
		5PM to 6PM	Sri. Y. Prasad Reddy	Structured Error Handling
9	29/07/2019	4PM to 5PM	Smt. B. Gouri	Module V: User control: Creation of user control
		5PM to 6PM	Smt. B. Gouri	Usage of user control
10	30/07/2019	4PM to 5PM	Smt. B. Gouri	Customer Control: Creation of customer control
		5PM to 6PM	Smt. B. Gouri	Usage of customer control
11	31/07/2019	4PM to 5PM	Sri. Y. Prasad Reddy	Gridview
		5PM to 6PM	Sri. Y. Prasad Reddy	Asp.net repeater control
12	01/08/2019	4PM to 5PM	Sri. Y. Prasad Reddy	File upload in asp.net
		5PM to 6PM	Sri. Y. Prasad Reddy	Asp.net security
13	02/08/2019	4PM to 5PM	Sri. Y. Prasad Reddy	Themes and master pages
		5PM to 6PM	Sri. Y. Prasad Reddy	App.config
14	03/08/2019	4PM to 5PM	Sri. Y. Prasad Reddy	Global.asax
		5PM to 6PM	Smt. B. Gouri	Module VI: web application using asp.net
15	05/08/2019	4PM to 5PM	Smt. B. Gouri	web application using asp.net
		5PM to 6PM	Smt. B. Gouri	Exam, Certificate Distribution and Vote of Thanks
16	06/08/2019	4PM to 6PM	Sri. S. Khaja Khizar Smt. B. Gouri Sri. Y. Prasad Reddy	

Coordinator

HoD

Dr. M. Sreeni[illegible]
M.E., Ph.D.
Professor & HOD CSE
K.S.R.M. College of Engineering
KADAPA - 516 003

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - AUTONOMOUS)

Kadapa, Andhra Pradesh, India - 516003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

An ISO 14001:2004 & 9001: 2015 Certified Institution

Department of Computer Science & Engineering

Certificate Course on .Net Technology

Attendance Sheet

S.No	Roll Num	Name of the Student	19/07/2019	20/07/2019	22/07/2019	23/07/2019	24/07/2019	25/07/2019	26/07/2019	27/07/2019	29/07/2019	30/07/2019	31/07/2019	01/08/2019	02/08/2019	03/08/2019	05/08/2019	06/08/2019
1	179Y1A0502	Adimulam Srinath	P	P	P	A	P	P	P	P	P	P	P	A	P	P	P	P
2	179Y1A0505	Ammireddy Nagamohan Reddy	P	P	A	P	P	P	P	P	A	P	P	P	P	P	P	P
3	179Y1A0506	Anandi Sai Charan Yadav	P	P	P	P	P	P	A	P	P	P	P	P	A	P	P	P
4	179Y1A0507	Avula Nagalakshmi	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
5	179Y1A0508	Balagani Sivachandra	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P
6	179Y1A0509	Balla Saikishor	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P
7	179Y1A0510	Bandaru Divya Sree	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
8	179Y1A0511	Bandepureddy Govardhan Reddy	P	P	P	A	P	P	P	P	P	A	P	P	P	P	P	P
9	179Y1A0512	Basireddu Gari Vikram Reddy	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P
10	179Y1A0513	Battina Manohar Reddy	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P
11	179Y1A0515	Bhumireddy Bhargavi	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P
12	179Y1A0516	B.Susmitha Priya	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P
13	179Y1A0517	Bojja Prabhanjan Reddy	P	P	P	P	P	A	P	P	P	P	A	P	P	P	P	P
14	179Y1A0518	Bokkasam Ganeshnaidu	P	P	P	P	A	P	P	P	P	P	P	P	P	P	A	P

15	179Y1A0519	Bonamsetty Reddyprasad	P	P	P	P	P	P	P	A	P	P	P	A	P	P	P	P
16	179Y1A0520	Bosigari Bharathi	P	P	P	P	P	P	P	P	A	P	P	P	P	A	P	P
17	179Y1A0522	C.Lohith Kumar	P	P	P	P	P	P	P	P	P	A	P	P	A	P	P	P
18	179Y1A0523	Chanda Harinadh Reddy	P	P	A	P	P	A	P	P	P	P	P	P	P	P	P	P
19	179Y1A0524	Chappidi Harsha Vardhan Reddy	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P
20	179Y1A0525	Chavva Sravani	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
21	179Y1A0526	C.V.S.Kowshiknath Reddy	P	P	P	A	P	P	P	P	P	P	P	P	A	P	P	P
22	179Y1A0527	Chenuru Brahmani	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
23	179Y1A0528	Cheppali Shaik Mahammed Aslam	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P
24	179Y1A0529	Chilakala Mahammadrafi	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P
25	179Y1A0531	Chinthala Akhila	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
26	179Y1A0532	D Mohammmad Sharuk	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P
27	179Y1A0533	Dabbera Sowmya	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
28	179Y1A0534	Daddanala Yogya Vignesh	P	P	A	P	P	P	P	P	P	P	P	P	P	A	P	P
29	179Y1A0537	D. Yogesh	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P
30	179Y1A0538	D Anusha	P	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P
31	179Y1A0539	Doddi Devananda	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P
32	179Y1A0540	D.Yasaswini	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P
33	179Y1A0543	D. Jyothsna	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P
34	179Y1A0544	E. Abhishek Kumar Sahu	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
35	179Y1A0547	G. Sudeesh Chandra Rao	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P
36	179Y1A0548	G. Haritha	P	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P
37	179Y1A0549	G. Vinod Kumar	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
38	179Y1A0552	G. Rajavardhan Reddy	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P
39	179Y1A0553	G. Sai Sujitha Reddy	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P
40	179Y1A0557	J. Haritha	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
41	179Y1A0558	J Sai Kumar	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P
42	179Y1A0560	K Kavitha Reddy	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
43	179Y1A0561	K Varshitha Reddy	P	P	P	P	P	P	P	P	A	P	P	P	P	P	P	P
44	179Y1A0562	Kambham Venkata Sai Kumar Reddy	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P
45	179Y1A0564	Kancharla Sasikala	P	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P
46	179Y1A0566	Kanchaveerla Harika	P	P	P	P	P	P	P	P	P	P	P	P	P	A	P	P

47	169Y1A0502	A. Sreekanth	P	P	A	P	P	P	P	P	A	P	P	P	P	P	P	P
48	169Y1A0509	B. Venkatesh	P	P	P	P	P	A	P	P	P	P	P	P	A	P	P	P
49	169Y1A0515	C. Bagya Sree	P	P	P	A	P	P	P	P	P	P	P	P	P	A	P	P
50	169Y1A0528	D. Jyoshna	P	P	P	P	P	P	P	P	P	A	P	P	P	A	P	P
51	169Y1A0532	G. Upendra	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
52	169Y1A0534	G. Krishna Kanth	P	P	P	P	P	P	P	P	P	P	P	P	A	P	A	P
53	169Y1A0546	K. Narasimha	P	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P
54	169Y1A0556	K. Supraja	P	P	P	P	P	P	P	P	P	P	A	P	P	P	P	P
55	169Y1A0559	K. Naresh	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
56	169Y1A0562	M. Srinivasulu	P	P	P	P	P	P	A	P	P	P	P	P	P	P	P	P
57	169Y1A0563	M. Ajay Kumar	P	P	P	P	P	P	P	P	P	P	P	A	P	A	P	P
58	169Y1A0564	M. Dinesh Reddy	P	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P
59	169Y1A0565	M. Suresh	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
60	169Y1A0566	M. Sai Yashwanth	P	P	A	P	P	P	P	P	P	P	P	P	P	P	P	P

Coordinator

HoD

Dr. M. Sreenivasulu,
M E, Ph. D.
Professor & HOD CSE
K.S.R.M. College of Engineering
K A D A P A - 516 003

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - Autonomous)

Kadapa, Andhra Pradesh, India– 516 003

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course on

.NET Technology

from 19-07-2019
to 06-08-2019

Venue : MB 210

(Lab : Web Technologies)

Coordinator : Mr. S. Khaja Khizar

Resource Person: Mrs. B. Gouri,
Mr. Y. Prasad Reddy

K.S.R.M. COLLEGE OF ENGINEERING

(AUTONOMOUS)

Pulivendala Road, Kadapa-516 005
Andhra Pradesh, India
Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.
An ISO 14001:2004 & 9001: 2015 Certified Institution

ACTIVITY REPORT

Certification Course

On

.Net Technology
19/07/2019 to 06/08/2019

Target Group	:	**Students**
Details of Participants	:	**60 Students**
Coordinator	:	**Mr. S. Khaja Khizar** **Asst. Prof, Dept. of CSE, KSRMCE**
Organizing Department	:	**Computer Science & Engineering**
Venue	:	**Web Technologies Lab (MB 210)**

Description: Certification course on ".Net Technology" was organized by Dept. of CSE from 19/07/2019 to 06/08/2019. Mr. S. Khaja Khizar acted as Course Coordinator and Mrs. B. Gouri and Mr. Y. Prasad Reddy acted as Resource Persons. The course is designed to provide complete knowledge of .Net Technology and to enhance the programming skills of the students by giving practical assignments to be done in labs. Thirty Seven hours course was successfully completed and participation certificates were provided to the participants.

Event Photos:

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - Autonomous)
Kadapa, Andhra Pradesh, India– 516 003
Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course on

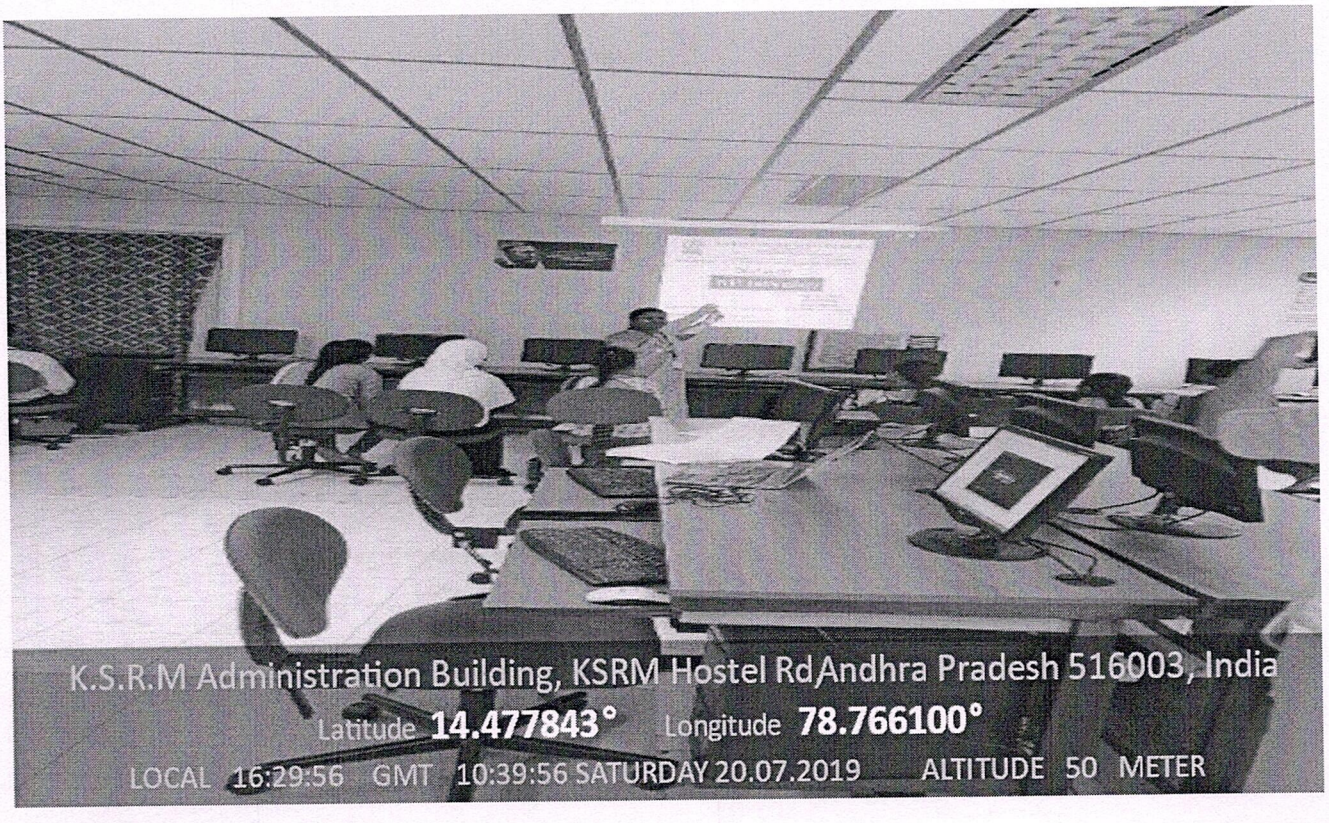
.NET Technology

from 19-07-2019
to 06-08-2019
Venue : MB 210
(Lab : Web Technologies)

Coordinator : Mr. S. Khaja Khizar

Resource Person: Mrs. B. Gouri,
Mr. Y. Prasad Reddy

Banner of the event



Resource person give brief overview about .net technology

Students participated in practical session

Certificate Distribution

Coordinator

HoD

Dr. M. Sreenivasulu,
M. E., Ph. D.
Professer & HOD CSE
K.S.R.M. College of Engineering

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - Autonomous)
Kadapa, Andhra Pradesh, India– 516 003
Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE OF PARTICIPATION

This is to certify that mr/miss. C. Lohith Kumar bearing roll no. 179Y1A0529 participated in a certification course on ".Net Technology" organized by department of computer science and engineering from 19 July to 06th Aug 2019.

COORDINATOR	HOD	PRINCIPAL

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - Autonomous)
Kadapa, Andhra Pradesh, India– 516 003
Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE OF PARTICIPATION

This is to certify that mr/miss. K. Kavitha Reddy bearing roll no. 179Y1A0560 participated in a certification course on ".Net Technology" organized by department of computer science and engineering from 19 July to 06th Aug 2019.

COORDINATOR	HOD	PRINCIPAL

K.S.R.M. COLLEGE OF ENGINEERING

(UGC - Autonomous)
Kadapa, Andhra Pradesh, India– 516 003
Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE OF PARTICIPATION

This is to certify that mr/miss. M. Suresh bearing roll no. 169Y1A0565 participated in a certification course on ".Net Technology" organized by department of computer science and engineering from 19 July to 06th Aug 2019.

COORDINATOR	HOD	PRINCIPAL

K.S.R.M. COLLEGE OF ENGINEERING

UGC - AUTONOMOUS

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

Kadapa, Andhra Pradesh, India– 516 003

FEEDBACK FORM

Certificate Course on ".Net Technology", from 19-07-2019 to 06-08-2019

Organized

by

Department of Computer Science & Engineering

NAME:

Roll No:

S.No	Feedback Item	Excellent	Very Good	Good	Average	Below Average
1	Organization of certificate course and session planning by resource person.					
2	Clarity in content delivery.					
3	Content is relevant and useful.					
4	Adequate opportunity to interact with resource person.					
5	Judicious mix of concepts, principles and practices.					
6	Assignments and tasks are interesting and challenging.					

Any suggestions for improvement.

Signature

K.S.R.M. COLLEGE OF ENGINEERING

UGC - AUTONOMOUS

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

Kadapa, Andhra Pradesh, India– 516 003

FEEDBACK FORM

Certificate Course on ".Net Technology", from 19-07-2019 to 06-08-2019

Organized

by

Department of Computer Science & Engineering

NAME: 179Y1A0515, B. Bhargavi

Roll No: 179Y1A0515

S.No	Feedback Item	Excellent	Very Good	Good	Average	Below Average
1	Organization of certificate course and session planning by resource person.	✓				
2	Clarity in content delivery.		✓			
3	Content is relevant and useful.	✓				
4	Adequate opportunity to interact with resource person.	✓				
5	Judicious mix of concepts, principles and practices.	✓				
6	Assignments and tasks are interesting and challenging.	✓				

Any suggestions for improvement.

B. Bhargavi

Signature

K.S.R.M. COLLEGE OF ENGINEERING

UGC - AUTONOMOUS

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

Kadapa, Andhra Pradesh, India– 516 003

FEEDBACK FORM

Certificate Course on ".Net Technology", from 19-07-2019 to 06-08-2019

Organized

by

Department of Computer Science & Engineering

NAME: K. Venkatasai kumarreddy

Roll No: 179Y1A0562

S.No	Feedback Item	Excellent	Very Good	Good	Average	Below Average
1	Organization of certificate course and session planning by resource person.	✓				
2	Clarity in content delivery.		✓			
3	Content is relevant and useful.	✓				
4	Adequate opportunity to interact with resource person.	✓				
5	Judicious mix of concepts, principles and practices.		✓			
6	Assignments and tasks are interesting and challenging.	✓				

Any suggestions for improvement.

Venkatasai

Signature

K.S.R.M. COLLEGE OF ENGINEERING

UGC - AUTONOMOUS

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

Kadapa, Andhra Pradesh, India– 516 003

FEEDBACK FORM

Certificate Course on ".Net Technology", from 19-07-2019 to 06-08-2019

Organized

by

Department of Computer Science & Engineering

NAME: K. Varshitha Reddy

Roll No: 179Y1A0561

S.No	Feedback Item	Excellent	Very Good	Good	Average	Below Average
1	Organization of certificate course and session planning by resource person.	✓				
2	Clarity in content delivery.			✓		
3	Content is relevant and useful.		✓			
4	Adequate opportunity to interact with resource person.	✓				
5	Judicious mix of concepts, principles and practices.			✓		
6	Assignments and tasks are interesting and challenging.	✓				

Any suggestions for improvement.

K. Varshitha

Signature

K.S.R.M. COLLEGE OF ENGINEERING

UGC - AUTONOMOUS

Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu.

Kadapa, Andhra Pradesh, India– 516 003

FEEDBACK FORM

Certificate Course on ".Net Technology", from 19-07-2019 to 06-08-2019

Organized

by

Department of Computer Science & Engineering

NAME: A. Sai charan Yadav.

Roll No: 179Y1A0506

S.No	Feedback Item	Excellent	Very Good	Good	Average	Below Average
1	Organization of certificate course and session planning by resource person.			✓		
2	Clarity in content delivery.	✓				
3	Content is relevant and useful.					
4	Adequate opportunity to interact with resource person.		✓		✓	
5	Judicious mix of concepts, principles and practices.			✓		
6	Assignments and tasks are interesting and challenging.	✓				

Any suggestions for improvement.

A. Sai charan Yadav.

Signature

Module1

Introduction to the .NET initiative and the .NET platform

The .NET Framework is a development framework that provides a new programming interface to Windows services and APIs, and integrates a number of technologies that emerged from Microsoft during the late 1990s. Microsoft announced the .NET initiative in July 2000. In April 2003, Version 1.1 of the integral .NET Framework was released. This book describes this updated version of the .NET Framework.

The .NET platform consists of four separate product groups:

Development tools and libraries

A set of languages, including C#, J#, and VB.NET; a set of development tools, including Visual Studio .NET; a comprehensive class library for building web services and web and Windows applications; as well as the Common Language Runtime (CLR). These components collectively form the largest part of the .NET Framework.

Web services

An offering of commercial web services, specifically the .NET Services initiative; for a fee, developers can use these services in building applications that require them.

Specialized servers

A set of .NET-enabled enterprise servers, including SQL Server, Exchange Server, BizTalk Server, and so on. These provide specialized functionality for relational data storage, email, and B2B commerce. Future versions of these products will increasingly support the .NET Framework.

Devices

New .NET-enabled, non-PC devices, from cell phones to game boxes.

Behind Microsoft .NET

While the main strategy of .NET is to enable software as a service, .NET is much more than that. In addition to embracing the Web, Microsoft .NET acknowledges and responds to the following trends within the software industry:

Distributed computing

Simplifies the development of robust client/server and multi-tier (*n*-tier) applications. Traditional distributed technologies require high vendor-affinity and are unable to interoperate with the Web. Microsoft .NET provides remoting and web services architectures that exploit open Internet standards, including the Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP) and WSOL.

Componentization

Simplifies the integration of software components developed by different vendors and supports development of distributed applications. The Component Object Model (COM) has brought reality to software plug-and-play, but COM component development and deployment are too complex. Microsoft .NET provides a simpler way to build and deploy components.

Enterprise services

Allow the development of scalable enterprise applications without writing code to manage transactions, security, or pooling. Microsoft .NET continues to support COM and component services, since these services greatly reduce the development time and effort required to build large-scale applications.

Web paradigm shifts

Over the past decade, web application development has shifted from connectivity (TCP/IP), to presentation (HTML), to programmability (XML and SOAP). A key goal of Microsoft .NET is to enable the sharing of functionality across the Web among different platforms, devices, and programming languages.

Maturity of IT industry

Lessons that the software industry has learned from developing large-scale enterprise and web applications. A commercial web application must support interoperability, scalability, availability, security, and manageability. Microsoft .NET facilitates all these goals.

Although these are the sources of many ideas embodied by Microsoft .NET, what's most notable about the platform is its use of open Internet standards (HTTP, XML, and SOAP) at its core to transmit information from one machine to another across the Internet. In fact, .NET provides bidirectional mapping between XML and objects. For example, a class can be expressed as an XML Schema Definition (XSD); an object can be converted to and from an XML buffer; a method can be specified using an XML format called Web Services Description Language (WSDL); and a method call can be expressed using an XML format called SOAP.

The .NET Platform

The Microsoft .NET platform consists of five main components, as shown in Figure 1-1. At the lowest layer lies the operating system (OS), which can be one of a variety of Windows platforms, including Windows XP, Windows 2000, Windows Server 2003, Windows ME, and Windows CE. As part of the .NET strategy, Microsoft has promised to deliver more .NET device software to facilitate a new generation of smart devices.

On top of the operating system is a collection of specialized server products that shortens the time required to develop large-scale business systems. These server products include Application Center, BizTalk Server, Commerce Server, Exchange Server, Host Integration Server, Internet Security and Acceleration Server, and SQL Server.

Since web services are highly reusable across the Web, Microsoft provides a number of building-block services (officially called .NET Services) that applications developers can use, for a fee. Two examples of .NET Services that Microsoft offers include .NET Passport and .NET Alerts. .NET Passport allows you to use a single username and password at all web sites that support Passport authentication. .NET Alerts allow .NET Alert providers, such as a business, to alert their consumers with important or up-to-the-minute information. Microsoft plans to add newer

services, such as calendar, directory, and search services. Third-party vendors are also creating new web services of their own.

The top layer of the .NET architecture is a development tool called Visual Studio .NET (VS.NET), which makes possible the rapid development of web services and other applications. A successor to Microsoft Visual Studio 6.0, VS.NET is an Integrated Development Environment (IDE) that supports four different languages and features such as cross-language debugging and the XML Schema Editor.

And at the center of .NET is the Microsoft .NET Framework—the main focus of this book. The .NET Framework is a development and runtime infrastructure that changes the development of business applications on the Windows platform. The .NET Framework includes the CLR and a common framework of classes that can be used by all .NET languages.

.NET Framework Design Goals

The Microsoft .NET Framework embodies design goals that are both practical and ambitious. In this section, we discuss the main design goals of the Microsoft .NET Framework, including support for components, language integration, application interoperation across the Web, simplified development and deployment, improved reliability, and greater security.

Component Infrastructure

Prior to the introduction of COM technology, developers had no standard way to integrate binary libraries without referring to or altering their source code. With the advent of COM, programmers were able to integrate binary components into their applications, similar to the way we can plug-and-play hardware components into our desktop PCs. Although COM was great, the grungy details of COM gave developers and administrators many headaches.

Although COM permits you to integrate binary components developed using any language, it does require you to obey the COM identity, lifetime, and binary layout rules. You must also write the plumbing code that is required to create a COM component, such as DllGetClassObject, CoRegisterClassObject, and others.

Realizing that these requirements result in frequent rewrites of similar code, .NET sets out to remove them. In the .NET world, all classes are ready to be reused at the binary level. You don't have to write extra plumbing code to support componentization in the .NET Framework. You simply write a .NET class, which then becomes a part of an assembly (to be discussed in Chapter 2) that inherently supports plug-and-play.

In addition to providing a framework to make development easier, .NET removes the pain of developing COM components. Specifically, .NET removes the use of the registry for component registration and eliminates the requirements for extraneous plumbing code found in all COM components, including code to support IUnknown, class factories, component lifetime, registration, dynamic binding, and others.

TIP

"Component" is a nasty word because one person may use it to refer to an object and another may use it to refer to a compiled binary module. To be consistent, this book uses the term "COM component" (or simply "component") to refer to a binary module, such as a DLL or an EXE.

Language Integration

COM supports *language independence*, which means that you can develop a COM component in any language you want. As long as your component meets all the rules spelled out in the COM specification, it can be instantiated and used by your applications. Although this supports binary reuse, it doesn't support *language integration*. In other words, you can't reuse the code in the COM components written by someone else; you can't extend a class hosted in the COM component; you can't catch exceptions thrown by code in the COM component; and so forth.

Microsoft .NET supports not only language independence, but also language integration. This means that you can inherit from classes, catch exceptions, and take advantage of polymorphism across different languages. The .NET Framework makes this possible with a specification called the Common Type System (CTS), which all .NET components must support. For example, everything in .NET is an object of a specific class that derives from the root class called System.Object. The CTS supports the general concepts of classes, interfaces, delegates (which support callbacks), reference types, and value types. The .NET base classes provide most

of the base system types, such as those that support integer, string, and file manipulation. Because every language compiler must meet a minimum set of rules stipulated by the Common Language Specification (CLS) and generate code to conform to the CTS, different .NET languages can be used in the same application. We will examine the CTS and CLS in Chapter 2.

Internet Interoperation

COM supports distributed computing through its Distributed COM (DCOM) wire protocol. A problem with DCOM is that it embeds the host TCP/IP address inside the Network Data Representation (NDR) buffer, such that it will not work through firewalls and Network Address Translation (NAT) software. In addition, the DCOM dynamic activation, protocol negotiation, and garbage collection facilities are proprietary, complex, and expensive. The solution is an open, simple, and lightweight protocol for distributed computing. The .NET Framework uses the industry-supported SOAP protocol, which is based on the widely accepted XML standards.

Simplified Development

If you have developed software for the Windows platforms since their inception, you have seen everything from the Windows APIs to the Microsoft Foundation Classes (MFC), the Active Template Library (ATL), the system COM interfaces, and countless other environments, such as Visual Interdev, Visual Basic, JScript, and other scripting languages. Each time you set out to develop something in a different compiler, you had to learn a new API or a class library, because there is no consistency or commonality among these different libraries or interfaces.

.NET solves this problem by providing a set of framework classes that every language uses. Such a framework removes the need for learning a new API each time you switch languages.

Simplified Deployment

Imagine this scenario: your Windows application, which uses three shared Dynamic Link Libraries (DLLs), works just fine for months, but stops working one day after you've installed another software package that overwrites the first DLL, does nothing to the second DLL, and adds an additional copy of the third DLL into a different directory. If you have ever encountered such a brutal—yet entirely possible—problem, you have entered DLL Hell. And if you ask a group of seasoned developers whether they have experienced DLL Hell, they will grimace at you

in disgust, not because of the question you've posed, but because they have indeed experienced the pain and suffering.

To avoid DLL Hell on Windows 2000 and subsequent Windows operating systems (at least for system DLLs), Windows 2000 stores system DLLs in a cache. If you install an application that overwrites system DLLs, Windows 2000 will overwrite the added system DLLs with the original versions from the cache.

Microsoft .NET further diminishes DLL Hell. In the .NET environment, your executable will use the shared DLL with which it was built. This is guaranteed, because a shared DLL must be registered against something similar to the Windows 2000 cache, called the Global Assembly Cache (GAC). In addition to this requirement, a shared DLL must have a unique hash value, public key, locale, and version number. Once you've met these requirements and registered your shared DLL in the GAC, its physical filename is no longer important. In other words, if you have two versions of a DLL that are both called *MyDll.dll*, both of them can live and execute on the same system without causing DLL Hell. This is possible because the executable that uses one of these DLLs is tightly bound to a specific version of the DLL during compilation.

In addition to eradicating DLL Hell, .NET also removes the need for component-related registry settings. A COM developer will tell you that half the challenge of learning COM is understanding the COM-specific registry entries for which the developer is responsible. Microsoft .NET stores all references and dependencies of .NET assemblies within a special section called a *manifest* (see Chapter 2). In addition, assemblies can be either private or shared. Private assemblies are found using logical paths or XML-based application configuration files, and public assemblies are registered in the GAC; in both cases, the system will find your dependencies at runtime. If they are missing, you get an exception telling you exactly what happened.

Finally, .NET brings back the concept of zero-impact installation and removal. This concept is the opposite of what you have to deal with in the world of COM. To set up a COM application, you have to register all your components after you have copied them over to your machine. If you fail to perform this step correctly, nothing will work and you'll end up pulling your hair out. Likewise, to uninstall the application, you should unregister your components (to remove the

registry entries) prior to deleting your files. Again, if you fail to perform this step correctly, you will leave remnants in the registry that will be forever extant.

Unlike COM, but like DOS, to set up an application in .NET, you simply xcopy your files from one directory on a CD to another directory on your machine, and the application will run automatically.[1] Similarly, you can just delete the directory to uninstall the application from your machine.

Reliability

There are many programming languages and platforms in the commercial software industry, but few of them attempt to provide both a reliable language and a robust runtime or infrastructure. The most successful language that we have seen in the commercial software industry is the Java™ language and the Java Virtual Machine™, which have brought the software-development community much satisfaction. Microsoft is positioning .NET as the next big thing.

Microsoft .NET requires type safety. Unlike C++, every class in .NET is derived from the mother of all classes, Object, which supports basic features such as returning a string representation of the object, indicating whether the object is equal to another, and so on. The CLR must recognize and verify types before they can be loaded and executed. This decreases the chances for rudimentary programming errors and prevents buffer overruns, which can be a security weakness.

Traditional programming languages don't provide a common error-handling mechanism. C++ and Java support exception handling, but many others leave you in the dust, forcing to invent your own error-handling facilities. Microsoft .NET supports exceptions in the CLR, providing a consistent error-handling mechanism. Put another way: exceptions work across all .NET-compatible languages.

When you program in C++, you must deallocate all heap-based objects that you have previously allocated. If you fail to do this, the allocated resources on your system will never be reclaimed even though they are no longer needed. And if this is a server application, it won't be robust because the accumulation of unused resources in memory will eventually bring down the system.

Similar to Java, the .NET runtime tracks and garbage-collects all allocated objects that are no longer needed.

Security

When developing applications in the old days of DOS, Microsoft developers cared little about security because their applications ran on a single desktop with a single thread of execution. As soon as developers started developing client and server applications, things got a bit complicated: multiple users might then have accessed the servers, and sensitive data might be exchanged between the client and the server. The problem became even more complex in the web environment, since you could unknowingly download and execute malicious applets on your machine.

To mitigate these problems, .NET provides a number of security features. Windows NT and Windows 2000 protect resources using access-control lists and security identities, but don't provide a security infrastructure to verify access to parts of an executable's code. Unlike traditional security support in which only access to the executable is protected, .NET goes further to protect access to specific parts of the executable code—this is known as *code access security*. For example, to take advantage of declarative security checks, you can prefix your method implementations with security attributes without having to write any code. To take advantage of imperative security checks, you write the code in your method to explicitly cause a security check. .NET provides other security features to make it harder to penetrate your applications and system.

.NET Framework

Now that you are familiar with the major goals of the .NET Framework, let's briefly examine its architecture. As you can see in Figure 1-2, the .NET Framework sits on top of the operating system, which can be a few different flavors of Windows and consists of a number of components (each of these components is discussed in greater detail starting with Chapter 4, as described in the Preface). .NET is essentially a system application that runs on Windows.

The most important component of the framework is the CLR. If you are a Java programmer, think of the CLR as the .NET equivalent of the Java Virtual Machine (JVM). If you don't know

Java, think of the CLR as the heart and soul of the .NET architecture. At a high level, the CLR activates objects, performs security checks on them, lays them out in memory, executes them, and garbage-collects them.

Conceptually, the CLR and the JVM are similar in that they are both runtime infrastructures that abstract the underlying platform differences. However, while the JVM officially supports only the Java language, the CLR supports any language that can be represented in its Common Intermediate Language (CIL). The JVM executes bytecode, so it can, in principle, support many languages, too. Unlike Java's bytecode, though, CIL is never interpreted. Another conceptual difference between the two infrastructures is that Java code runs on any platform with a JVM, whereas .NET code runs only on platforms that support the CLR. In April, 2003, the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC) recognized a functional subset of the CLR, known as the Common Language Interface (CLI), as an international standard. This development, initiated by Microsoft and developed by ECMA International, a European standards organization, opens the way for third parties to implement their own versions of the CLR on other platforms, such as Linux or Mac OS X. For information on third-party and open source projects working to implement the ISO/IEC CLI and C# specifications, see Appendix A.

In Figure 1-2, the layer on top of the CLR is a set of framework base classes. This set of classes is similar to the set of classes found in STL, MFC, ATL, or Java. These classes support rudimentary input and output functionality, string manipulation, security management, network communications, thread management, text management, reflection functionality, collections functionality, as well as other functions.

On top of the framework base classes is a set of classes that extend the base classes to support data management and XML manipulation. These classes, called ADO.NET, support persistent data management—data that is stored on backend databases. Alongside the data classes, the .NET Framework supports a number of classes to let you manipulate XML data and perform XML searching and XML translations.

Classes in three different technologies (including web services, Web Forms, and Windows Forms) extend the framework base classes and the data and XML classes. Web services include a

number of classes that support the development of lightweight distributed components, which work even in the face of firewalls and NAT software. These components support plug-and-play across the Internet, because web services employ standard HTTP and SOAP.

Web Forms, the key technology behind ASP.NET, include a number of classes that allow you to rapidly develop web Graphical User Interface (GUI) applications. If you're currently developing web applications with Visual Interdev, you can think of Web Forms as a facility that allows you to develop web GUIs using the same drag-and-drop approach as if you were developing the GUIs in Visual Basic. Simply drag-and-drop controls onto your Web Form, double-click on a control, and write the code to respond to the associated event.

Windows Forms support a set of classes that allow you to develop native Windows GUI applications. You can think of these classes collectively as a much better version of the MFC in C++ because they support easier and more powerful GUI development and provide a common, consistent interface that can be used in all languages.

Madule2

Code Management

The Microsoft .NET Framework has two main components: the Common Language Runtime (CLR) and the .NET Framework class library. The CLR, which is the foundation of the .NET Framework, is a negotiator that manages code at execution time. The concept of code management is a fundamental principle of the CLR, and applications that the CLR manages are called "managed applications" or "managed code."

When it compiles a managed application, the CLR's language compiler (e.g., Visual Basic .NET, C#) translates source code into Microsoft intermediate language (MSIL), which is a CPU-independent set of instructions that the language compiler then efficiently converts to native code. Before you can execute the code, you must convert MSIL to CPU-specific code, usually by using a just-in-time (JIT) compiler. JIT compilation takes into account that the a specific CPU might not ever call some code during execution. Therefore, rather than using processing time and memory to convert all the MSIL to native code, the JIT compiler converts the MSIL on an as-needed basis during execution. When the JIT compiler converts MSIL, the compiler stores the resulting native code so that it's accessible for subsequent calls. Because the CLR supplies JIT

compilers for each computer architecture that CLR supports, you can use them to compile and execute the same set of MSIL code on any supported architecture.

Managed-application compilation raises an intriguing question. If you can use JIT compilers to compile and execute MSIL on any supported architecture, what's preventing Microsoft (or any other software vendor) from creating a CLR that runs on IBM or Sun Microsystems machines, for example? My company has many enterprise clients with huge investments in IBM and Sun hardware. These investments are costly because of their software licensing. But what if you could run Windows (or IIS), whose licensing model is dramatically less expensive, on those machines? The possibilities are mind-boggling!

Class libraries, the second .NET Framework component, provide standard functionality such as I/O, string manipulation, security management, network communications, thread management, and text management. Data classes let you use standard interfaces to manipulate Microsoft SQL Server and XML data. For example, you can use XML Web-service classes to develop distributed components that work across firewalls and Network Address Translation (NAT) software. Web Forms include classes that let you develop Web GUI applications and provide a common development interface across all the languages that the .NET Framework supports.

Microsoft's successful 2-year .NET Framework and Visual Studio .NET beta program was the largest in the company's history. But with so little time and so much to do, many companies didn't even consider .NET let alone install it in beta environments—until its ship date a few weeks ago. And let's face it: A large contingent of companies won't consider installing and leveraging the .NET Framework for months, possibly years. Many companies have decided that the software's deployment costs (even though the product is free), the costs of the learning curve (on both the development and the infrastructure sides of the house), and the fear of putting new software into production are too significant. However, some companies put the software into production in beta forms—within or outside Microsoft's Early Adopter Program (EAP). Companies such as the Home Shopping Network (HSN), Merrill Lynch, and Verizon Communications have stepped on stages across the country at various Microsoft events and

boldly proclaimed their successes with .NET. And now that .NET Framework is shipping, many technologists aren't so reticent to install it and put it through its paces.

This week should be interesting. Microsoft's official Visual Studio .NET launch event is scheduled for February 13 in San Francisco, and Microsoft Chairman and Chief Software Architect Bill Gates will present the keynote address. Local Microsoft offices around the world are planning similar events on or around the February 13 date; some will show the keynote address in a live or taped format. I'll be attending the San Francisco event, so look for my report in the next installment.

Module3

Object oriented programming concepts

Here I will explain oops concepts in c# with examples or object oriented programming concepts in c#.net with real time examples or oops or object oriented programming in asp.net with example.

Now let's start learning about OOPS concepts in c#.net with real time examples

OOPS Concepts

Class:

It is a collection of objects.

Object:

It is a real time entity.
An object can be considered a "**thing**" that can perform a set of **related** activities. The set of activities that the object performs defines the object's behavior. For example, the hand can grip something or a **Student** (object) can give the name or address. In pure **OOP** terms an object is an instance of a class