

## Lesson Plan for Even semester

**Govt. Polytechnic, Jhajjar**

**Name of the Faculty:** VIVEK DAHIYA  
**Discipline:** Computer Engineering  
**Semester:** 4th  
**Subject:** Microprocessors and Peripheral Devices  
**Lesson Plan Duration:** 15 weeks (from January, 2020 to April, 2020)  
**Work Load (Lecture/ Practical) per week (in hours):** Lect- 03 Pract-06

Week	Theory		Practical	
	Lecture day	Topic (including assignment/ test)	Practical day	Topic
1 <sup>st</sup>	1 <sup>st</sup>	Typical organization of a microcomputer system.	1 <sup>st</sup>	A brief history of Microprocessor. How Microprocessor works and its various features.
	2 <sup>nd</sup>	Functions of its various blocks.		
	3 <sup>rd</sup> th	Microprocessor and its evolution.		
	4	Function and impact of microprocessor on modern society.		
2 <sup>nd</sup>	5 <sup>th</sup>	Concept of Bus.	2 <sup>nd</sup>	Familiarization of different keys of 8085 microprocessor kit.
	6 <sup>th</sup>	Bus organization of 8085.		
	7 <sup>th</sup>	Functional block diagram of 8085 and function of each block.		
	8 <sup>th</sup>	Pin details of 8085 and related signals.		
3 <sup>rd</sup>	9 <sup>th</sup>	Demultiplexing of address/data bus generation of read/write control signals.	3 <sup>rd</sup>	Familiarization of memory map of 8085.

	10 <sup>th</sup>	Steps to execute a stored Programme.		
	11 <sup>th</sup>	Instruction cycle, machine cycle and T-states.		
	12 <sup>th</sup>	Fetch and execute cycle.		
4 <sup>th</sup>	13 <sup>th</sup>	Revision and assignments	4 <sup>th</sup>	Steps to enter data/program on 8085 kit.
	14 <sup>th</sup>	TEST		
	15 <sup>th</sup>	Brief idea of machine and assembly languages, Machines and Mnemonic codes, Instruction format and Addressing mode.		
	16 <sup>th</sup>	Identification of instructions as to which addressing mode they belong.		
5 <sup>th</sup>	17 <sup>th</sup>	Concept of Instruction set and Explanation of the instructions of the following groups of instruction set.	5 <sup>th</sup>	Steps to modify data/program on 8085 kit
	18 <sup>th</sup>	Data transfer group		
	19 <sup>th</sup>	Revision and assignments		
	20 <sup>th</sup>	TEST		
6 <sup>th</sup>	21 <sup>st</sup>	Arithmetic Group, Logic Group	6 <sup>th</sup>	Steps to execute a programme on 8085 kit.
	22 <sup>nd</sup>	Stack, I/O and Machine Control Group		
	23 <sup>th</sup>	Revision and assignments		
	24 <sup>th</sup>	TEST		

7 <sup>th</sup>	25 <sup>th</sup>	Programming exercises in assembly language	7 <sup>th</sup>	Writing and execution of ALP for addition of two 8 bit numbers.
	26 <sup>th</sup>	Concept of memory mapping		
	27 <sup>th</sup>	Partitioning of total memory space		
	28 <sup>th</sup>	Revision and assignments		
	29 <sup>th</sup>	Revision and assignments		
8 <sup>th</sup>	30 <sup>th</sup>	TEST	8 <sup>th</sup>	Writing and execution of ALP for subtraction of two 8 bit numbers.
	31 <sup>st</sup>	Address decoding		
	32 <sup>nd</sup>	Concept of peripheral mapped I/O		
	33 <sup>rd</sup>	Concept of memory mapped I/O		
9 <sup>th</sup>	34 <sup>th</sup>	Interfacing of memory mapped I/O devices	9 <sup>th</sup>	Writing and execution of ALP for multiplication and division of two 8 bit numbers.
	35 <sup>th</sup>	Revision and assignments		
	36 <sup>th</sup>	TEST		
	37 <sup>th</sup>	Concept of interrupt		
10 <sup>th</sup>	38 <sup>th</sup>	Maskable and non-maskable	10 <sup>th</sup>	Writing and execution of ALP for arranging 10 numbers in ascending/descending order.
	39 <sup>th</sup>	Edge triggered and level triggered interrupts		
	40 <sup>th</sup>	Software interrupt		
	41 <sup>st</sup>	Restart interrupts and its use		
11 <sup>th</sup>	42 <sup>nd</sup>	Various hardware interrupts of 8085	11 <sup>th</sup>	Writing and execution of ALP for 0 to 9 BCD counters (up/down counter according to choice stored in memory).
	43 <sup>rd</sup>	Servicing interrupts		
	44 <sup>th</sup>	Revision and assignments		
	45 <sup>th</sup>	extending interrupt		

12 <sup>th</sup>		system	12 <sup>th</sup>	Interfacing exercise on 8255 like LED display control and 8253 programmable interval timer.
	46 <sup>th</sup>	Concept of programmed I/O operations		
	47 <sup>th</sup>	synchronous data transfer, asynchronous data		
		transfer (hand shaking)		
	48 <sup>th</sup>	Interrupt driven data transfer		
13 <sup>th</sup>	49 <sup>th</sup>	DMA	13 <sup>th</sup>	Interfacing exercise on 8279 programmable KB/display interface like to display the hex code of key pressed on display.
	50 <sup>th</sup>	Serial output data, Serial input data		
	51 <sup>st</sup>	8255 PPI		
	52 <sup>nd</sup>	8253 PIT		
14 <sup>th</sup>	53 <sup>rd</sup>	8257 / 8237 DMA controller,	14 <sup>th</sup>	Use of 8085 emulator for hardware testing.
	54 <sup>th</sup>	Programmable KB/Display Interface,		
	55 <sup>th</sup>	8251 Communication Interface Adapter.		
	56 <sup>th</sup>	Revision and assignments		
15 <sup>th</sup>	57 <sup>th</sup>	Revision and assignments	15 <sup>th</sup>	Revision and Problem Solving.
	58 <sup>th</sup>	Revision and assignments		
	59 <sup>th</sup>	Revision and assignments		
	60 <sup>th</sup>	Revision and assignments		

## LESSON PLAN

Name of faculty : Yashvir Singh Lecturer (G.P. Jhajjar)

Discipline : Computer Engineering

Subject : OOPJ (4<sup>th</sup>sem)

Lesson plan duration : 15 weeks (from January, 2020 to

April, 2020) Workload (lecture/practical) per week: Lectures-

03, practicals-06

week	Theory		Practical	
	Lecture Day	Topic (including assignment/test)	Practical	Topic
1 <sup>st</sup>	1 <sup>st</sup>	Introduction and Features : Fundamentals of Object oriented programming – procedure oriented programming Vs. object Oriented programming (OOP.) Object oriented programming concepts – Classes, object, object reference	1 <sup>st</sup>	Program of basic OOP in java.
	2 <sup>nd</sup>	<i>Abstraction, encapsulation Inheritance,</i>		
	3 <sup>rd</sup>	<i>Inheritance, polymorphism, Introduction of eclipse (IDE) for developing programs in Java</i>	2 <sup>nd</sup>	Program of basic OOP in java.
2 <sup>nd</sup>	1 <sup>ST</sup>	<b>Language Constructs</b> : Review of constructs of C used in JAVA :	1 <sup>st</sup>	Consider we have a Class of Cars under which Santro Xing, Alto and Wagon R represents individual Objects. In this context each Car Object will have its own, Model, Year of Manufact., Color, Top Speed, etc. which form Properties of the Car class and the associated actions i.e., object Functions like Create(), Sold(), display() form the Methods of Car Class. Use this class to create another class Company that tracks the modelist create.
	2 <sup>ND</sup>	data types, increment and decrement operators		
	3 <sup>RD</sup>	Relational and logical operators, if else then clause		
			2 <sup>nd</sup>	Software Engineers, Module Lead, Technical Lead, Project Lead, Project Manager,

				Program Manager, Directors all are the employees of the company but their work, perks, roles, responsibilities differs. Create the Employee base class would provide the common behavior so of all types of employee and also some behaviors properties that all employee must have for that company. Also include search method
3 <sup>rd</sup>	1 <sup>st</sup>	Conditional expressions ,input using scanner class	1 <sup>st</sup>	Suppose the Airport person also want to maintain records of the arrival and departure of the planes. Create a class Airport that has data like name, id, and address.
	2 <sup>nd</sup>	Input using scanner class and output statement,		
	3 <sup>rd</sup>	Output statement Loops,	2 <sup>nd</sup>	Practice of practical's.
4 <sup>th</sup>	1 <sup>st</sup>	Switch case	1 <sup>st</sup>	. Create a whole menu driven hospital management system using concept of OOP like classes, inheritance. Include information about the following: a. Patient - name, registration id, age, disease, etc. b. Staff - id, name, designation, salary, etc. c.
	2 <sup>nd</sup>	Arrays		
	3 <sup>rd</sup>	Methods	2 <sup>nd</sup>	Practice of practical.
5 <sup>TH</sup>	1 <sup>st</sup>	<b>Classes and Objects:</b> Creation, accessing class members	1 <sup>st</sup>	Create a class called Musician should contain three methods string(), wind() and perc(). Each of these methods should initialize a string array to contain the following instruments: veena, guitar, sitar, sarod and mandolin under string() - flute, clarinet, saxophone, nadaswaram and piccolo under wind() - tabla, mridangam, bangos, drums and tambour under perc()
	2 <sup>nd</sup>	Private Vs Public Vs Protected Vs Default		
	3 <sup>rd</sup>	Revision /Test	2 <sup>nd</sup>	Practice of practical.
6 <sup>TH</sup>	1 <sup>st</sup>	<b>Constructors</b>	1 <sup>st</sup>	Write three derived classes inheriting functionality of

	2 <sup>nd</sup>	Object & Object Reference		baseclassperson(shouldhaveamemberfunctionthatastasktoenternameandage)andwithaddeuniquefeaturesofstudent,andemployee,andfunctionalitytoassign,changeanddeleterecordsofstudentandemployee.
	3 <sup>rd</sup>	Object & Object Reference	2 <sup>nd</sup>	Practice of practical's.
7 <sup>th</sup>	1 <sup>st</sup>	Inheritance: Definition of inheritance, protected data,	1 <sup>st</sup>	Usingtheconceptofmultipleinheritancecreateclasses:Shape,Circle,Square,Cube,Sphere,Cylinder. YourclassesmayonlyhavetheclassvariablespecifiedinthetablebelowandthethodsAreaand/orVolumetooutputtheirareaand/orvolume.
	2 <sup>nd</sup>	Private data, public data,		
	3 <sup>rd</sup>	Constructor chaining, order of invocation	2 <sup>nd</sup>	Write a program to create Class Person.
8 <sup>th</sup>	1 <sup>st</sup>	Order of invocation, types of inheritance,	1 <sup>st</sup>	To create class STUDENT inherit from Person
	2 <sup>nd</sup>	Single inheritance Multilevel inheritance,	2 <sup>nd</sup>	To create class Instructor inherits from Person.
	3 <sup>rd</sup>	Hierarchical inheritance		
9 <sup>th</sup>	1 <sup>st</sup>	Hierarchical inheritance Hybrid inheritance	1 <sup>st</sup>	To create class Instructor inherit from Person.
	2 <sup>nd</sup>	Hybrid inheritance		
	3 <sup>rd</sup>	Polymorphism: Method & constructor overloading,	2 <sup>nd</sup>	Writetheclassdefinitions,theconstructors,setmethods,getmethodsandforallclasses.
10 <sup>th</sup>	1 <sup>st</sup>	Method overriding	1 <sup>st</sup>	Writetheclassdefinitions,theconstructors,setmethods,getmethodsandforallclasses.
	2 <sup>nd</sup>	up-casting, down-casting		
	3 <sup>rd</sup>	Revision /Test	2 <sup>nd</sup>	
11 <sup>th</sup>	1 <sup>st</sup>	Abstract class& Interface	1 <sup>st</sup>	Write the classdefinitions,theconstructors,setmethods,getmethodsandforallclasses.
	2 <sup>nd</sup>	Abstract class &Interface		
	3 <sup>rd</sup>	Implementation of multiple inheritance through	2 <sup>nd</sup>	Write the class definitions, the constructors ,set

		interface		Methods, get methods and for all classes.
12 <sup>th</sup>	1 <sup>st</sup>	Implementation of multiple inheritance through interface	1 <sup>st</sup>	Write the class definitions ,the constructors ,set methods, get Methods and for all classes.
	2 <sup>nd</sup>	Implementation of multiple inheritance through interface	2 <sup>nd</sup>	9.OldMacDonaldhadafarman dseveral typesofanimals.Every animalsharedcertaincharacte ristics:theyhadatype(suchasc ow,chickorpig)andeachmade asound(moo,cluck).AnInterfa cedefinesthose thingsrequire dtobeananimalonthefarm.De finenewclassesfortheOldMac Donald that implement the Animal and Farm class. Create array of object of animal to define the different types of animalinthefarm.Alsocreatea ppropriatemethodstogetands ettheproperties
	3 <sup>rd</sup>	Revision of Abstract class & Interface and discuss problems		
13 <sup>th</sup>	1 <sup>st</sup>	Exception Handling:	1 <sup>st</sup>	10.WriteaprogramwithStude ntasabstractclassandcreated eriveclassesEngineering,Med icineandSciencefrombasecla ssStudent.Create theobjectso fthederivedclassesandproce ssthemandaccessthemusing arrayofpointerof typebasecla ssStudent.
	2 <sup>nd</sup>	Implementation of keywords like try and catch		
	3 <sup>rd</sup>	Implementation of keywords like finally, throw &throws.		Practice of practical..
14 <sup>th</sup>	1 <sup>st</sup>	Importance of exception handling in practical implementation of live projects	1 <sup>st</sup>	Revision of practical.
	2 <sup>nd</sup>	Importance of exception handling in practical implementation of live projects		
	3 <sup>rd</sup>	Revision and problems	2 <sup>nd</sup>	Revision of practical.
15 <sup>th</sup>	1 <sup>st</sup>	Revision and problems	1 <sup>st</sup>	Revision of practical.
	2 <sup>nd</sup>	Revision and problems		
	3 <sup>rd</sup>	Revision /Test	2 <sup>nd</sup>	Revision of practical.



# Lesson Plan

Name of the Faculty : Shakti Raj Singh, Programmer

Discipline : Computer Engg.

Semester : 4<sup>th</sup>

Subject : Data Structure using C

Lesson plan duration : 15 weeks (from 9<sup>th</sup> January, 2020 to 30<sup>th</sup> April, 2020) Theory-3hr, Practical-6hrs

Week	The		Practical	
	Lecture Day	Topic (including assignments /tests)	Practical Day	Topic
1 <sup>st</sup> Week	1 <sup>st</sup>	Problem solving concept , Top-down and bottom-up design, structured programming	1 <sup>st</sup>	Exercise of C Program
			2 <sup>nd</sup>	Exercise of C Program
	2 <sup>nd</sup>	Concept of data type, variables and constants	1 <sup>st</sup>	Exercise of C Program /Revision/Practice Session
	3 <sup>rd</sup>	Introduction to data Structure( Linear, Non Linear, Primitive, Non Primitive))	2 <sup>nd</sup>	Exercise of C Program /Revision/Practice Session
Week 2	1 <sup>st</sup>	Concept of Data Structure (Array, Linked List, Stack, Queue, Trees, Graphs)	1 <sup>st</sup>	Exercise of C Program
			2 <sup>nd</sup>	Exercise of C Program
	2 <sup>nd</sup>	Concept of Arrays	1 <sup>st</sup>	Program regarding Array/Revision/Practice Session
	3 <sup>rd</sup>	One dimensional Array, Two Dimensional Array: Representation of Two dimensional Array ( Base address, LB, UB )	2 <sup>nd</sup>	Program regarding Array/Revision/Practice Session
Week 3	1 <sup>st</sup>	Operational on Arrays with Algorithms (inserting, deleting )	1 <sup>st</sup>	Program regarding Array
	2 <sup>nd</sup>	Operational on Arrays with Algorithms (Searching, Traversing	2 <sup>nd</sup>	Program regarding Array
	3 <sup>rd</sup>	Introduction to linked list and double linked list, Representation of Linked list in Memory	1 <sup>st</sup>	Program regarding Array/Revision/Practice Session
2 <sup>nd</sup>			Program regarding Array/Revision/Practice Session	
Week 4	1 <sup>st</sup>	Describe and Comparison between Linked list and Array	1 <sup>st</sup>	Program of Matrices
	2 <sup>nd</sup>	Traversing and Searching Linked List	2 <sup>nd</sup>	Program of Matrices
	3 <sup>rd</sup>	Insertion and deletion into Linked list	1 <sup>st</sup>	Program of Matrices/ Revision/ Practice Session
			2 <sup>nd</sup>	Program of Matrices/Revision/Practice Session
Week 5	1 <sup>st</sup>	Application of Linked List and Explain Doubly Linked List	1 <sup>st</sup>	Program of addition of two Matrices using function
	2 <sup>nd</sup>	Traversing, Insertion and deletion into doubly Linked List	2 <sup>nd</sup>	Program of addition of two Matrices using function
	3 <sup>rd</sup>	Introduction to Stack, Representation of Stacks With Array and Linked list	1 <sup>st</sup>	Program of addition of two Matrices using function/Revision/PracticeSession
			2 <sup>nd</sup>	Program of addition of two Matrices using function/Revision/PracticeSession
Week 6	1 <sup>st</sup>	Implementation of Stacks	1 <sup>st</sup>	Program of inserting and deleting elements in array

		Converting Infix to Post Fix Notation)	2 <sup>nd</sup>	Program of addition of two Matrices using function/Revision/PracticeSession
	3 <sup>rd</sup>	Evaluation of Post fix Notation and Tower of Hanoi	1 <sup>st</sup>	Program of inserting and deleting elements in array /Revision/Practice Session
			2 <sup>nd</sup>	Program of inserting and deleting elements in array /Revision/Practice Session
Week 7	1 <sup>st</sup>	Recursion : Concept and Comparison between recursion and Iteration	1 <sup>st</sup>	Program of Push and POP Operation in stack
	2 <sup>nd</sup>	Introduction of Queues and Implementation of queues ( array and <u>Linked list with algorithm</u> )	2 <sup>nd</sup>	Program of Push and POP Operation in stack
	3 <sup>rd</sup>	Introduction of Queues and Implementation of queues ( array and <u>Linked list with algorithm</u> )	1 <sup>st</sup>	Program of Push and POP Operation in stack /Revision/Practice Session
			2 <sup>nd</sup>	Program of Push and POP Operation in stack /Revision/Practice Session
Week 8	1 <sup>st</sup>	Explain Circular Queues and De-Queues	1 <sup>st</sup>	Program of Conversion from infix notation
	2 <sup>nd</sup>	Introduction of Trees and Concept of Binary Trees	2 <sup>nd</sup>	Program of Conversion from infix notation
	3 <sup>rd</sup>	Explain Complete and Extended Binary Tree	1 <sup>st</sup>	Program of Conversion from infix notation/Revision/PracticeSession
			2 <sup>nd</sup>	Program of Conversion from infix notation/Revision/Practice Session
Week 9	1 <sup>st</sup>	Concept of representation of Binary Tree	1 <sup>st</sup>	Program of the Factorial of given number using recursion
	2 <sup>nd</sup>	Concept of representation of balanced Binary Tree	2 <sup>nd</sup>	Program of the Factorial of given number using recursion
	3 <sup>rd</sup>	Explain Traversing Binary Trees (Pre Order, Post Order and In Order)	1 <sup>st</sup>	Program of the Factorial of given number using recursion /Revision/Practice Session
			2 <sup>nd</sup>	Program of the Factorial of given number using recursion /Revision/Practice Session
Week 10	1 <sup>st</sup>	Explain Searching, inserting and deleting in binary seary trees	1 <sup>st</sup>	Insertion and Deletion of elements in Queue and Circular Queue using Pointer
	2 <sup>nd</sup>	Explain Searching, inserting and deleting in binary seary trees	2 <sup>nd</sup>	Insertion and Deletion of elements in Queue and Circular Queue using Pointer
	3 <sup>rd</sup>	Problems Solution	1 <sup>st</sup>	Insertion and Deletion of elements in Queue and Circular Queue using Pointer /Revision/Practice Session
			2 <sup>nd</sup>	Insertion and Deletion of elements in Queue and Circular Queue using Pointer /Revision/Practice Session
	1 <sup>st</sup>	Test	1 <sup>st</sup>	Insertion and Deletion of elements

Week 11	2 <sup>nd</sup>	Problems Solution		in Linked List and doubly Linked list
			2 <sup>nd</sup>	Insertion and Deletion of elements in Linked List and doubly Linked list
	3 <sup>rd</sup>	Previous topic Explain	1 <sup>st</sup>	Insertion and Deletion of elements in Linked List and doubly Linked list/Revision/Practice Session
			2 <sup>nd</sup>	Insertion and Deletion of elements in Linked List and doubly Linked list/Revision/Practice Session
Week 12	1 <sup>st</sup>	Introduction of Sorting and Searching	1 <sup>st</sup>	Program of Linear Search procedures to search an element in given list
	2 <sup>nd</sup>	Search algorithm( Linear and Binary)	2 <sup>nd</sup>	Program of Linear Search procedures to search an element in given list
	3 <sup>rd</sup>	Search algorithm( Linear and Binary)	1 <sup>st</sup>	Program of Binary Search procedures to search an element in given list/Revision/Practice Session
			2 <sup>nd</sup>	Program of Binary Search procedures to search an element in given list/Revision/Practice Session
Week 13	1 <sup>st</sup>	Concept and uses of Sorting	1 <sup>st</sup>	Previous Problems solution
			2 <sup>nd</sup>	Previous Problems solution
	2 <sup>nd</sup>	Sorting Algorithm (Bubble sort )	1 <sup>st</sup>	Previous Problems solution /Revision/Practice Session
	3 <sup>rd</sup>	Sorting Algorithm (Insertion sort )		
		2 <sup>nd</sup>	Previous Problems solution /Revision/Practice Session	
Week 14	1 <sup>st</sup>	Sorting Algorithm (Selection sort )	1 <sup>st</sup>	Program of Bubble Sort
	2 <sup>nd</sup>	Sorting Algorithm (Merge Sort )	2 <sup>nd</sup>	Program of Bubble Sort
	3 <sup>rd</sup>	Sorting Algorithm (Radix sort ) & Sorting Algorithm ( Heap Sort )	1 <sup>st</sup>	Program of Bubble Sort/Revision/Practice Session
			2 <sup>nd</sup>	Program of Bubble Sort/Revision/Practice Session
Week 15	1 <sup>st</sup>	Problems Solution	1 <sup>st</sup>	Program of Selection Sort
	2 <sup>nd</sup>	Problems solution	2 <sup>nd</sup>	Program of Selection Sort
	3 <sup>rd</sup>	Test	1 <sup>st</sup>	Program of Selection Sort /Revision/Practice Session
			2 <sup>nd</sup>	Program of Selection Sort /Revision/Practice Session

## LESSON PLAN

Name of the Faculty:- Reenu  
 Subject Computer organization  
 Semester 4<sup>th</sup>  
 Session Jan to may 2020

WEEK NO.	THEORY DAY	TOPICS COVERED	PRACTICAL
1	1	Introduction of computer hardware	N/A
	2	CPU organization	N/A
	3	Three address,two address,one address zero	N/A
2	1	RISC Instruction	N/A
	2	Addressing modes: Immediate,register,direct	N/A
	3	CPU Design: Microprog.vs hard wired	N/A
3	1	Reduced instruction set computer	N/A
	2	CISC characteristics	N/A
	3	Revision & class test	N/A
4	1	Introduction to memory	N/A
	2	Memory Hirerachy	N/A
	3	RAM and ROM chips	N/A
5	1	Memory connection to CPU	N/A
	2	Auxillary Memory	N/A
	3	Cache memory	N/A
6	1	Virtual memory	N/A
	2	Memory Management hardware	N/A
	3	Revision calss test	N/A
7	1	Introduction to O/I organization	N/A
	2	Functions of BIOS and test	N/A
	3	Test and Initialization, configuring the system	N/A
8	1	Introduction to modes of data transfer	N/A
	2	Explain Programmed I/O	N/A
	3	Assignment questions revision	N/A
9	1	Various types of interrupts	N/A
	2	DMA data transfer	N/A
	3	Revision & class test	N/A
10	1	Introduction to Architecture of Computer	N/A
	2	Multi processor systems	N/A
	3	Forms of parallel processing	N/A
11	1	introduction to Multiprocessor	N/A
	2	Multi processor systems in detail	N/A
	3	revision -class test	N/A
12	1	Forms of parallel processing	N/A
	2	Parallel processing and pipelines,	N/A
	3	Basic charactersteristics	N/A
13	1	Interconnection network	N/A
	2	Time shared bus	N/A
	3	System bus	N/A
14	1	Multi ports	N/A
	2	Cross bar switch	N/A
	3	Multi stage	N/A
15	1	Switching networks	N/A
	2	Hyper cube structures.	N/A
	3	Revision & class test	N/A

