



JEPPIAAR INSTITUTE OF TECHNOLOGY
“Self-Belief | Self Discipline | Self Respect”



QUESTION BANK

Regulation : 2017

Year/Semester : III

Semester : 06

Batch : 2017-2021

**DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING**

Vision of the Institution

Jeppiaar Institute of Technology aspires to provide technical education in futuristic technologies with the perspective of innovative, industrial and social application for the betterment of humanity

Mission of the Institution

M1: To produce competent and disciplined high-quality professionals with the practical skills necessary to excel as innovative professionals and entrepreneurs for the benefit of the society.

M2: To improve the quality of education through excellence in teaching and learning, research, leadership and by promoting the principles of scientific analysis, and creative thinking.

M3: To provide excellent infrastructure, serene and stimulating environment that is most conducive to learning.

M4: To strive for productive partnership between the Industry and the Institute for research and development in the emerging fields and creating opportunities for employability.

M5: To serve the global community by instilling ethics, values and life skills among the students needed to enrich their lives.

DEPARTMENT VISION

To produce Engineers with visionary knowledge in the field of Computer Science and Engineering through scientific and practical education in stance of inventive, modern and communal purpose for the improvement of society.

DEPARTMENT MISSION

M1: Devise students for technical and operational excellence, upgrade them as competent engineers and entrepreneurs for country's development.

M2: Develop the standard for higher studies and perpetual learning through creative and critical thinking for the effective use of emerging technologies with a supportive infrastructure.

M3: Involve in a constructive, team-oriented environment and transfer knowledge to balance the industry-institute interaction.

M4: Enrich students with professional integrity and ethical standards that will make them deal social challenges successfully in their life.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: To support students with substantial knowledge for developing and resolving mathematical, scientific and engineering problems.

PEO 2: To provide students with adequate training and opportunities to work as a collaborator with informative and administrative qualities.

PEO 3: To motivate students for extensive learning to prepare them for graduate studies, R&D and competitive exams.

PEO 4: To cater students with industrial exposure in an endeavour to succeed in the emerging cutting edge technologies.

PEO 5: To shape students with principled values and to follow the code of ethics in social and professional life.

PROGRAM SPECIFIC OUTCOMES (PSOS)

PSO 1 : Students are able to analyse, design, implement and test any software with the programming and testing skills they have acquired.

PSO 2: Students are able to design and develop algorithms for real time problems, scientific and business applications through analytical, logical and problems solving skills.

PSO 3: Students are able to provide security solution for network components and data storage and management which will enable them to work efficiently in the industry.

BLOOM'S TAXONOMY

Definition:

- A theory to identify cognitive levels (Levels of thinking)
- Represents the full range of cognitive functions.

Objectives:

- To classify educational learning objectives into levels of complexity and specificity. The classification covers the learning objectives in cognitive, affective and sensory domains.
- To structure curriculum learning objectives, assessments and activities.

Levels in Bloom's Taxonomy:

- **BTL 1 – Remember** - The learner is able to recall, restate and remember learned information.
- **BTL 2 – Understand** - The learner grasps the meaning of information by interpreting and translating what has been learned.
- **BTL 3 – Apply** - The learner makes use of information in a context similar to the one in which it was learned.
- **BTL 4 – Analyze** - The learner breaks learned information into its parts to best understand that information.
- **BTL 5 – Evaluate** - The learner makes decisions based on in-depth reflection, criticism and assessment.
- **BTL 6 – Create** - The learner creates new ideas and information using what has been previously learned.

TABLE OF CONTENT

CS8651-INTERNET PROGRAMMING		
Unit No.	Topic	Page No.
	Syllabus	1.1
I	Website Basics	1.3
II	Client-side programming	1.8
III	Server-side programming	1.13
IV	PHP and XML	1.17
V	Introduction to AJAX and web services	1.21
CS8691-ARTIFICIAL INTELLIGENCE		
	Syllabus	2.1
I	Introduction	2.2
II	Problem solving methods	2.6
III	Knowledge representation	2.9
IV	Software agents	2.15
V	Generating functions	2.20
CS8601-MOBILE COMPUTING		
	Syllabus	3.1
I	Introduction	3.3
II	Mobile internet protocol and transport layer	3.13
III	Mobile Telecommunication System	3.23
IV	Mobile Ad-hoc networks	3.33
V	Mobile platforms and applications	3.43
CS8602-COMPILER DESIGN		
	Syllabus	4.1
I	Introduction to compilers	4.3
II	Lexical analysis	4.14
III	Syntax analysis	4.22
IV	Syntax directed translation & run time environment	4.33
V	Code optimization and zcode generation	4.47

CS8603-DISTRIBUTED SYSTEMS		
	Syllabus	5.1
I	Introduction	5.3
II	Message ordering & snapshots	5.13
III	Distributed mutex & deadlock	5.21
IV	Recovery & consensus	5.28
V	P2p & distributed shared memory	5.33
IT8076-SOFTWARE TESTING		
	Syllabus	6.1
I	Introduction	6.2
II	Test case design Strategies	6.10
III	Levels of Testing	6.16
IV	Test Management	6.24
V	Test Automation	6.30

CS8651	INTERNET PROGRAMMING	L TPC
		3 0 03

OBJECTIVES

- To understand different Internet Technologies.
- To learn java-specific web services architecture

UNIT I WEBSITE BASICS**9**

Web Essentials: Clients, Servers and Communication – The Internet – Basic Internet protocols – World wide web – HTTP Request Message – HTTP Response Message – Web Clients – Web Servers – HTML5 – Tables – Lists – Image – HTML5 control elements – Semantic elements – Drag and Drop – Audio – Video controls - CSS3 – Inline, embedded and external style sheets – Rule cascading – Inheritance – Backgrounds – Border Images – Colors – Shadows – Text – Transformations – Transitions – Animations.

UNIT II CLIENT SIDE PROGRAMMING**9**

Java Script: An introduction to JavaScript–JavaScript DOM Model-Date and Objects,- Regular Expressions- Exception Handling-Validation-Built-in objects-Event Handling- DHTML with JavaScript- JSON introduction – Syntax – Function Files – Http Request – SQL.

UNIT III SERVER SIDE PROGRAMMING**9**

Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies- Installing and Configuring Apache Tomcat Web Server- DATABASE CONNECTIVITY: JDBC perspectives, JDBC program example - JSP: Understanding Java Server Pages-JSP Standard Tag Library (JSTL)-Creating HTML forms by embedding JSP code.

UNIT IV PHP AND XML**9**

An introduction to PHP: PHP- Using PHP- Variables- Program control- Built-in functions- Form Validation- Regular Expressions - File handling – Cookies - Connecting to Database.XML: Basic XML- Document Type Definition- XML Schema DOM and Presenting XML, XML Parsers and Validation, XSL and XSLT Transformation, News Feed(RSS and ATOM).

UNIT V INTRODUCTION TO AJAX and WEB SERVICES**9**

AJAX: Ajax Client Server Architecture-XML Http Request Object-Call Back Methods; Web Services: Introduction- Java web services Basics – Creating, Publishing, Testing and Describing a Web services (WSDL)-Consuming a web service, Database Driven web service from an application – SOAP.

TOTAL 45 PERIODS

OUTCOMES:

Upon completion of the course, students will be able to:

- Construct a basic website using HTML and Cascading Style Sheets.
- Build dynamic web page with validation using Java Script objects and by applying different event handling mechanisms.
- Develop server side programs using Servlets and JSP.
- Construct simple web pages in PHP and to represent data in XML format.
- Use AJAX and web services to develop interactive web applications

TEXT BOOKS:

1. Deitel and Deitel and Nieto, -Internet and World Wide Web - How to Program , Prentice Hall, 5th Edition, 2011.

REFERENCES:

1. Stephen Wynkoop and John Burke -Running a Perfect Website, QUE, 2nd Edition,1999.
2. Chris Bates, Web Programming – Building Intranet Applications, 3rd Edition, Wiley Publications, 2009.
3. Jeffrey C and Jackson, -Web Technologies A Computer Science Perspective, Pearson Education, 2011.
4. Gopalan N.P. and AkilandeswariJ., -Web Technologyl, Prentice Hall of India, 2011.
UttamK.Roy, -Web Technologiesl, Oxford University Press, 2011.

Subject Code: CS8651 **Year/Semester: III /06**
Subject Name: INTERNET PROGRAMMING **Subject Handler:S.SUDHAMERCY**

UNIT 1 - WEBSITE BASICS, HTML 5, CSS 3, WEB 2.0

Web Essentials: Clients, Servers and Communication – The Internet – Basic Internet protocols – World wide web – HTTP Request Message – HTTP Response Message – Web Clients – Web Servers – HTML5 – Tables – Lists – Image – HTML5 control elements – Semantic elements – Drag and Drop – Audio – Video controls - CSS3 – Inline, embedded and external style sheets – Rule cascading – Inheritance – Backgrounds – Border Images – Colors – Shadows – Text – Transformations – Transitions – Animations.

PART * A

Q.NO	QUESTIONS
1.	<p>How to write comment statements in HTML? BT1</p> <p>The comments in HTML can be denoted as follows - <!--It is a comment statements--> There should not be a space between angular bracket and exclamation mark. This comment is beginning with <!-- and ending with-->. There should not be any character such as-- inside the comment.</p>
2.	<p>State the three types of XHTML DTDs along with their usage. BTL2</p> <p>There are three types of XHTML DTDs and those along with their uses are as given below-</p> <p>XHTML 1.0 Strict: When we want a clean markup code then this type of dtd is used.</p> <p>XHTML 1.0 Transitional: When we want to use some html features in the existing XHML document.</p> <p>XHTML 1.0 Frameset: When want to make use of frames in the XHTML document .</p>
3.	<p>How will you create a password field in a HTML form? BTL2</p> <pre><form name="form1"> <input type="password" name="pwd" value=""></form></pre>
4.	<p>Differentiate client side and server side scripts? BTL 2</p> <p>Client-side Environment</p> <p>The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.</p> <p>The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off. When</p>

	<p>this is the case a message usually pops up to alert the user when script is attempting to run.</p> <p>Server-side Environment</p> <p>The server-side environment that runs a scripting language is a web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.</p>
5.	<p>What is cell padding and cell spacing attributes? BTL2</p> <p>The cell padding allows to have some space between the contents of each cell and its borders. The distance between each is called cell spacing.</p>
6.	<p>What are style sheets? List the ways of including style information in a HTML document. [NOV/DEC 2018]BTL 1</p> <p>Style sheets are collections of style information that are applied to plain text. Style information includes font attributes such as type size, special effects (bold,italic,underline),color and alignment. Style sheets also provide broader formatting instructions by specifying values for quantities such as line spacing and left and right margins. The ways of including style sheet are : Internal ,External and Inline stylesheet.</p>
7.	<p>List down the ways of including style information in document.APR/MAY 2019</p> <p>BTL 1</p> <p>External Styles -Style information is read from a separate file that is specified in the <LINK> tag <STYLE> and </STYLE> tags.</p>
8.	<p>Discuss the core syntax of CSS. BTL 2</p> <p>The selector points to the HTML element you want to style.</p> <p>The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.</p> <p>A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.</p> <p>Selector{ property:value; property:value;}</p>

9.	Give some advantages of using cascading style sheets (CSS) [APRIL/MAY 2018] . BTL 2	A simple mechanism for adding style(such as fonts,colors, or spacing) to web documents. Multiple levels of CSS can be used to allow selective overriding of styles.
11.	Explain with an example for inline style sheet.	BTL 3 <pre><html> <head> <style type="text/css"> h3{ color:green; } </style> </head> <body> <h3>This page contains an inline style sheet </h3> </body> <html></pre>
12.	Define Normal Flow Box Layout in CSS.	BTL 1 Normal Flow Box Layout is the default layout.
13.	What is meant by canvas in HTML?	BTL 1 The HTML <canvas> element is used to draw graphics on a web page. The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.
14.	How external style sheet is useful in web page design?	BTL1 When we use external style sheet then the style is defined in one file and the actual content of the web page are defined in another file. Hence if we want to change the style of presentation of web page then we can simply modify the style sheet file alone.
15.	What is API -Application Program Interface?	BTL 1 A set of routines, protocols, and tools for building software applications. A good API makes it easier to develop a program by providing all the building blocks. A programmer puts the blocks together.
16.	Write a note on Internet Information Server (IIS).	BTL 1 Microsoft's Web server that runs on Windows NT platforms, In fact, IIS comes bundled with Windows NT 4.0. Because IIS is tightly integrated with the operating system, it is relatively easy to administer.
17.	Define CGI -Common Gateway Interface.	BTL 1 A static member class is a static member of a class. Like any other static method, a specification for transferring information between a World Wide Web server and a CGI

	program. CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language ,including C, Perl, Java, or Visual Basic.	
18.	Define URL [APRIL/MAY 2018]	BTL 1
	URL (Uniform Resource Locator): It is a specification for identifying an object such as a file, newsgroup, CGI program or e-mail address by indicating the exact location on the internet.	
19.	State the difference between internet and intranet. [NOV / DEC 2016] [APRIL/MAY 2019] [APRIL/MAY 2018]	BTL 1
	There's one major distinction between an intranet and the Internet: The Internet is an open, public space, while an intranet is designed to be a private space. An intranet may be accessible from the Internet, but as a rule it's protected by a password and accessible only to employees or other authorized users.	
20	Define Rich Internet Applications. [MAY/JUN 2016] [NOV / DEC 2016] [APR / MAY 2017] A rich Internet application (RIA) is a Web application designed to deliver the same features and functions normally associated with desktop applications. RIAs generally split the processing across the Internet/network divide by locating the user interface and related activity and capability on the client side, and the data manipulation and operation on the application server side.	BTL 1
	PART * B	
1	(i)List and explain any four HTML elements in detail. (7M) (ii)Classify the types of lists supported by HTML and describe them in detail.(6M) Answer: Page: 12 - Technical Publications (i). List,Table ,Form ,Image HTML Elements – Use -[2 M] ,Syntax –[2M] and example –[3M] (ii) Ordered List,Unordered List,Definition List with example. [6M]	BTL 4
2	2. Briefly discuss about (i) HTML frames. (6M) (ii) Table tags. (7M) Answer: Page: 14 - Technical Publications (i). Need of Frames [2M] with example document [4M] Table tags <table>,<tr>,<td>,<th> with example. [Use-2M,Syntax-2M,Program –[3M]	BTL 2
3	3 Create a HTML document for a company home page and give details. [13M]	BTL 2

	Answer: Page : 30 - Technical Publications It should include all form elements for company home page. [Html Page – 13M]
4	Create a website using HTML for a “Library management system”. Your website should have a home page which helps the user to navigate to various pages like student membership, books catalog, transactions and search pages. [13M] BTL 6 Answer: Page : 23 - Technical Publications The above document should include all form elements, anchor tags for Library management system. [Html Page -(10M), Design -(3M)]
5	(i)List and explain in detail the various selector strings. (7M) (ii). Discuss the features of cascading style sheets. (6M) BTL 2 Answer: Page : 46 - Technical Publications (i)Class selector,Id Selector,Simple selector,Pseudo random classes.[7M] (ii). Features – Separation,Persistent,consistent look and feel,consistent appearance [6M]
PART C	
1	Create a website using HTML for an “Online Shopping System”. Your website should have a home page which helps the user to navigate to various pages like product category, product details ,payment details , search options pages . (13M) [APRIL/MAY 2019] BTL4 Answer: Page : 34 - Technical Publications Create a html page with form tags with all elements.[13M]
2	Describe the CSS box model in detail. (7M) BTL 2 (ii) List and explain in details about any four types of selector strings.(6M) Answer: Page : 52 - Technical Publications Margin,padding,content area,Edges with diagram (7M) (ii)Class ,Id,Pseudorandom,simple selectors with examples (6M)
3	(i) Express a CSS rule which adds background images and indentation.(7M) BTL 2 (ii) Define external style sheet with an example.(6M) Answer: Page : 62 - Technical Publications (i)background-image:url(“submarine.jpg”) with program (7M) (ii)A.Defintion Applying style to more than one web pages (2M) B.Syntax : <link href=”ex2.css”> to be included for external style sheet.(4M)

UNIT II CLIENT SIDE PROGRAMMING

Java Script: An introduction to JavaScript–JavaScript DOM Model–Date and Objects,- Regular Expressions- Exception Handling–Validation–Built-in objects–Event Handling– DHTML with JavaScript- JSON introduction – Syntax – Function Files – Http Request – SQL.

PART A

1	What is JavaScript statement? Give an example. <p>The statement in Javascript is referred as “Assignment statement” which will assign values to variables. The assignment statement in JavaScript is very much similar to C. For example :</p> <p>Sum+=10 Sum=sum+10 can be written in JavaScript.</p>	BTL 1								
2	List out the objects used in JavaScript with its purpose. <p>BTL 1</p> <p>Math,Boolean,Date,Number,String.</p>									
3	List the different methods defined in document and window object of JavaScript[nov/dec 2018]. <p>BTL1</p> <p>Alert Box, Prompt box and Confirm box are the methods used in window object.</p>									
4	What are global functions in Javascript?NOV/DEC 2017 <p>BTL 1</p> <p>The global functions are the top level functions in javascript that are independent of any specific object. These functions are built in objects of Some intrinsic attributes are listed in the following table</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">INSTRINSIC Attributes</th><th style="text-align: left; padding: 5px;">Meaning</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;">Onblur</td><td style="padding: 5px;">This event is for losing the focus</td></tr> <tr> <td style="padding: 5px;">Onchange</td><td style="padding: 5px;">On occurrence of some change this event occurs.</td></tr> <tr> <td style="padding: 5px;">Onclick</td><td style="padding: 5px;">When user clicks the mouse button this event occurs.</td></tr> </tbody> </table>	INSTRINSIC Attributes	Meaning	Onblur	This event is for losing the focus	Onchange	On occurrence of some change this event occurs.	Onclick	When user clicks the mouse button this event occurs.	
INSTRINSIC Attributes	Meaning									
Onblur	This event is for losing the focus									
Onchange	On occurrence of some change this event occurs.									
Onclick	When user clicks the mouse button this event occurs.									
5.	Write array creation in JavaScript with example. <p>BTL 1</p> <p>Javascript the array can be created using Array object. Suppose, we want to create an array of 10 elements then we can write, var ar=new Array(10);</p>									

	<p>Using new operator we can allocate the memory dynamically for the arrays. In the brackets the size of an array is mentioned and the var ar denotes the name of the array. Thus by the above sentence an array ar will be created in which we can store 10 elements at the most. Sometimes the above statement can be written like this var ar;</p> <pre>ar=new Array(10);</pre>
6.	<p>What is a DOM? [NOV/DEC 2017] BTL 1 A Document Object Model is an application programming interface that defines how javascript programs can access and manipulate the HTML</p>
7	<p>What is an Event? [NOV/DEC 2015] BTL 1 An event in a browser is an occurrence of potential interest. Example events are the mouse moving over an element. a mouse button being clicked, or a key being pressed. Each type of event has an abbreviated name associated with it.</p>
8	<p>Differentiate Dynamic document with Static document? BTL 1</p> <ul style="list-style-type: none"> An HTML document that contains scripting is called a dynamic document whereas a simple HTML document without scripting is known as a static document.
9	<p>What are the stages in a Servlet life cycle? BTL 1 1.init() service() 3.destroy()</p>
10	<p>What is Event listener in DOM? BTL 1 An event listener is a function that takes a single argument that is an instance of Event. A call to the addEventListener () method on a node object associates an event listener with a type of event occurring on that node.</p>
11	<p>What is a Session? BTL 1 A collection of HTTP requests all associated with a single session ID is known as a session. Each HTTP request is examined by the server to see if it contains a special identifier known as a session ID..</p>
12	<p>What are HttpServletRequest and HttpServletResponse? BTL 1 from the javax.servlet.http package. The HttpServletRequest enables the servlet to read data from the HTTP request and HttpServletResponse enables the servlet to write the data to the HTTP response.</p>
13	<p>What are the actions involved in a doGet()method? BTL 1 1.Set the HTTP Content-Type header of the response. 2.Obtain a PrintWriter object from the HttpServletResponse parameter object. 3.Output a valid HTML document to the PrintWriter object. 4.Close the PrintWriter object.</p>

14	How to create arrays in Javascript? We can declare an array like this Var scripts = new Array(); We can add elements to this array like this <code>scripts[0] = "PHP";</code> <code>scripts[1] = "ASP";</code> <code>scripts[2] = "JavaScript"; scripts[3]</code> <code>= "HTML";</code>	BTL 1
15	What are the primitive data types in javascript	BTL 1
PART B		
1	Create a website using HTML for a “Library management system”. Your website should have a home page which helps the user to navigate to various pages like student membership, books catalog, transactions and search pages. [13M] Answer: Page : 23 - Technical Publications The above document should include all form elements, anchor tags for Library management system. [Html Page -(10M), Design -(3M)]	
2	(i) Describe how do you use JavaScript for form validation? Develop a complete application that would include functions to validate the user data. (8M) [APRIL/MAY 2018] (ii) Write short notes on JavaScript built-in objects. (5M) BTL 2 Answer: Page : 35 - Technical Publications (i) Form elements should be validated for any application. [Validation -8M] Math object, String Object, Date, Number, Boolean with example.[Syntax-(2M), example-(3M)]	
3	Explain objects and arrays in JavaScript with suitable example. (13M) Answer Page: 38 - Technical Publications BTL2 Arrays and Objects: A .Definition (2M) Arrays is a collection of Similar data types and object is a collection of properties.	

	B. Syntax(2M) Example program.(9M)
4	<p>(i)Write JavaScript to find sum of first ‘n’ even number and display the result. Get the value of n from user. [APRIL/MAY 2019] (7M)</p> <p>(ii) Write JavaScript to find factorial of a given number.(6M)</p> <p>Answer: Page : 38 - Technical Publications</p> <p>Program :</p> <pre><script type="text/javascript"> Var f=prompt("enter the number",""); for(i=1;i<=5;i++) { f=f*i; } document.writeln("fact of 5 is"+f); </script> [13M]</pre>
5	<p>Write a Javascript program to delete the rollno property from the following object. Also print the object before and after deleting the property. Sample object: var student = { name: “Santhosh Ravy”, class: “VI”, rollno:29}; (7M)</p> <p>(i). Write a JavaScript program to search a date (MM/DD/YYYY) within a string.(6M) BTL 3</p> <p>Answer: Page : 36 - Technical Publications</p> <p>(i) A. Object Creation (2M)</p> <p>B. Deletion of property example. (2M)</p> <p>C. Example program (3M)</p> <p>Program (6M)</p>
PART C	
1	<p>Explain how you use JavaScript for form validation. Develop a login form application that would include functions to validate the user data.(9M)</p> <p>[APRIL/MAY 2018] (ii)Write short notes on JavaScript built in objects.(6M) BTL 4</p> <p>Answer: Page : 55 - Technical Publications</p> <p>(i) Login screen page with HTML and Javascript. [HTML Page -(5M),Javascript-(4M)]</p> <p>(ii) Math, String, Boolean, Date, Number[6M]</p>
2	<p>i) Create a JavaScript program to find the given number is odd or even.(7M)</p> <p>ii) Explain in detail about the CSS.(8M) BTL 4</p>

	<p>Answer: Page : 41 - Technical Publications</p> <p>(i) if($\text{num} \% 2 == 0$) {document.writeln("even"); } else {document.writeln("odd");} [7M]</p> <p>(ii) Box Models(3M) Types of CSS(3M) Advantages of CSS(2M)</p>
3	<p>Write a Javascript program to delete the rollno property from the following object. Also print the object before and after deleting the property. Sample object: var student = { name: "Santhosh Ravy", class: "VI", rollno:29}; (7M)</p> <p>(ii). Write a JavaScript program to search a date (MM/DD/YYYY) within a string.(6M) BTL 3</p> <p>Answer: Page : 36 - Technical Publications</p> <p>(i) A. Object Creation (2M) B. Deletion of property example. (2M) C. Example program (3M) Program (6M)</p>
4	<p>i) Describe how do you use JavaScript for form validation? Develop a complete BTL 2 application that would include functions to validate the user data.(8M)</p> <p>(ii) Write short notes on JavaScript built-in objects.(5M)</p> <p>Answer: Page : 35 - Technical Publications</p> <p>(i) Form elements should be validated for any application. [Validation -8M] Math object, String Object, Date, Number, Boolean with example. [Syntax-(2M), example-(3M)]</p>

UNIT III SERVER SIDE PROGRAMMING

Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies- Installing and Configuring Apache Tomcat Web Server-
DATABASE CONNECTIVITY: JDBC perspectives, JDBC program example - **JSP:** Understanding Java Server Pages-JSP Standard Tag Library (JSTL)-Creating HTML forms by embedding JSP code.

PART A

1	Define servlet	BTL1
2	Define the use of ServletContext object. The servletContext object can be used to communicate with the servlet container. When one ServletContext object is created then it can be accessed by all the servlets in the web container. For instance – if you want to share some data to all the servlets then you can share this data using the ServletContext object.	BTL1
3	List the use of cookies. [NOV/DEC 2018] BTL1 Cookies are some little information that can be left on your computer by the other computer when we access an internet. The information is stored in the cookies using the name value pair.	
4	What is JSP? Write two main usages of it. [NOV/DEC 2018] BTL1 1. Java Servlets contains pure Java Code and there is no mixing of HTML tags 2. Java Servlets execute faster than JSP Servlet can act as a model-view and controller all at the same time. It can handle different dynamic pages	
5	What are the advantages of event delegation model? The event-delegation model has two advantages over the event-inheritance model. First, Enables event handling to be handled by objects other than the ones that	BTL1

	<p>generate the events (or their containers). This allows a clean separation between a component's design and its use.</p> <p>The other advantage of the event-delegation model is that it performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to repeatedly process unhandled events, as is the case of</p>															
6	<p>What are the life cycle methods of servlets?</p> <p>server can invoke the service for particular HTTP request. Finally the server unloads the servlet from the memory using destroy() method.</p>															
7	<p>Classify the different types of directive in JSP?</p> <p>Directives are JSP elements that provide global information about an entire JSP page.</p> <p>The page directive defines information that will be globally available for that Java Server Page,</p> <ol style="list-style-type: none"> 1. language 2. extends 3. import 4. session 5. buffer 6. contenttype 															
8	<p>Compare JSP and servlet [NOV/DEC 2018] BTL1</p> <table border="1"> <thead> <tr> <th>SL.NO</th> <th>JSP</th> <th>SERVLET</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>JSP is an HTML in which JAVA code is embedded</td> <td>Servlets are basically JAVA program in which the HTML is embedded</td> </tr> <tr> <td>2</td> <td>JSP is a scripting language which generate the dynamic web page</td> <td>Servlets are java program that are compiled to generate dynamic web contents</td> </tr> <tr> <td>3</td> <td>In Model View Controller architecture(MVC) the JSP acts as a view</td> <td>In Model View Controller architecture(MVC) the Servlets acts as a Controller</td> </tr> <tr> <td>4</td> <td>JSP makes use of custom tags that can call the Java beans directly</td> <td>Servlets does not support for the custom tags</td> </tr> </tbody> </table>	SL.NO	JSP	SERVLET	1	JSP is an HTML in which JAVA code is embedded	Servlets are basically JAVA program in which the HTML is embedded	2	JSP is a scripting language which generate the dynamic web page	Servlets are java program that are compiled to generate dynamic web contents	3	In Model View Controller architecture(MVC) the JSP acts as a view	In Model View Controller architecture(MVC) the Servlets acts as a Controller	4	JSP makes use of custom tags that can call the Java beans directly	Servlets does not support for the custom tags
SL.NO	JSP	SERVLET														
1	JSP is an HTML in which JAVA code is embedded	Servlets are basically JAVA program in which the HTML is embedded														
2	JSP is a scripting language which generate the dynamic web page	Servlets are java program that are compiled to generate dynamic web contents														
3	In Model View Controller architecture(MVC) the JSP acts as a view	In Model View Controller architecture(MVC) the Servlets acts as a Controller														
4	JSP makes use of custom tags that can call the Java beans directly	Servlets does not support for the custom tags														
9	<p>How sessions are handled in servlets . BTL1</p> <p>Through cookies ,url rewriting and hidden form fields.</p>															

10	State JDBC BTL1 Java Database Connectivity or in short JDBC is a technology that enables the java program to manipulate data stored into the database.
11	What are the various database connectivity supported in Java? BTL Type1 driver, Type-II Driver, Type-III , Type IV Driver
12	Define socket. What are the classes used for socket programming? BTL1 The socket is a software abstraction used to represent the terminals of a connection between two machines or processes. 1.ServerSocket 2.Datagram Socket 3.Multicast Socket 4.Secure sockets
13	Mention the purpose of using URL class. BTL1 URL is the acronym for Uniform Resource Locator. It is a reference (an address) to a resource on the Internet. You provide URLs to your favorite Web browser so that it can locate files on the Internet in the same way that you provide addresses on letters so that the post office can locate your correspondents.
14	What do you mean by listen () method? BTL1 listen() marks the socket referred to by sockfd as a passive socket , that is, as a socket that will be used to accept incoming connection requests using accept
15	Give the need of Registry objects. BTL1 A remote object registry is a bootstrap naming service that is used by RMI servers on the same host to bind remote objects to names. Clients on local and remote hosts can then look up remote objects and make remote method invocations
16	Define the Callable Statement in JDBC? BTL1 The callable statement is used to represent stored procedures.These are similar to prepared statements. Eg: select * from students where rollno <25.
17	Write the steps involved in creating a JavaBean. BTL1 <ol style="list-style-type: none"> 1. Step 1: Put this source code into a file named "SimpleBean.java" ... 2. Step 2: Compile the file: ... 3. Step 3: Create a manifest file, named "manifest.tmp": ... 4. Step 4: Create the JAR file, named "SimpleBean.jar": ... Start the Bean Box.

18	How can you create JDBC statements? BTL1 A Statement is an interface that represents a SQL statement. You execute Statement objects, and they generate Result Set objects, which is a table of data representing a database result set. You need a Connection object to create a Statement object.
19	What is meant by prepared statement in JDBC BTL1 The prepared statement in JDBC is a precompiled statement. It is specified using a placeholder ?. eg: select * from students where name=?
20	Can the InetAddress class functionality to detect the IP Addresses, be handled using URL class? If yes, Explain? BTL1 Yes the InetAddress class is used to detect the ip address. Eg : InetAddress ia=InetAddress.getLocalHost();
PART B	
1	(i) Describe how JDBC works. (6M) BTL2 (ii) Show the various JDBC driver types in detail. (7M) Answer: Page : 87 - Technical Publications (i) JDBC Architecture (6M) Types of JDBC Driver –Type1,2,3,4 Drivers (4M) Diagram(3M)
2	(i) Write a client server JSP program to find simple interest and display the result in the client. (8M) (ii) Define the JSP tag libraries. (5M) BTL3 (i) JSP Program –(8M) Core,xml,sql are tag libraries –(5M)
3	(i) Explain the steps involved to create JDBC connectivity. List the advantages of JDBC. (7M) (ii) Explain the various methods used in ResultSet interface. (6M) BTL2 Answer: Page : 35 - Technical Publications (i) class.forName(""),connobj.getConnection("uname",pwd","database name"); Stmt.createStatement(), Resultset rs=stmt.executeQuery(sql query). (7M) hasElements(),next(),getInt(),getFloat() (6M)
4	(i) Discuss about servlet life cycle with example. [NOV/DEC 2018] (5M) (ii) Discuss database connectivity with Servlet to display student marks. (8M) [APRIL/MAY 2018] BTL2 (i) init(),service(),destroy() –(5M)

	Unit-III [Refer JDBC] – (8M)
5	<p>(i) Illustrate in detail with an example the dynamic content generation by servlet. (7M)</p> <p>(ii) Illustrate how java servlets perform session handling.(6M) BTL3</p> <p>(i) Dynamic content generation Program (7M) By Cookies (3M) URL Rewriting (3M)</p>
PART C	
1	<p>Explain with an example how a Java application can access a database using JDBC (15M)</p> <p>Answer: Page : 112 - Technical Publications BTL 2</p> <p>JDBC Connectivity – Select – for accessing [3M] Program[12M]</p>
2	<p>Create a Java Program those queries for student information from a database. The program must also facilities insertion, deletion, and updation of student details into the database(15M) BTL 5</p> <p>Answer: Page :120 - Technical Publications JDBC Program –Insertion [5M] JDBC Program deletion [5M] JDBC Program Updation[5M]</p>
3	<p>Discuss the concept of exception handling with an application of your choice. Write necessary code snippets. (15) MAY/JUNE 2017 BTL 3</p> <p>Key points:</p> <ol style="list-style-type: none"> 1. Creation of Exception class (5) 2. Sample code includes try and catch block (6) 3. Catching exceptions (2) 4. Sample code for different exceptions(2) <p>Answer: Page No. 299 in Herbert Schildt</p>
UNIT 4 - UNIT IV PHP and XML	
<p>An introduction to PHP: PHP- Using PHP- Variables- Program control- Built-in functions- Form Validation- Regular Expressions - File handling – Cookies - Connecting to Database.XML: Basic XML- Document Type Definition- XML Schema DOM and Presenting XML, XML Parsers and Validation, XSL and XSLT Transformation, News Feed (RSS and ATOM).</p>	
PART A	

1	What is PHP? [NOV / DEC 2016] [NOV/DEC 2017] [NOV/DEC 2015] BTL1 Hypertext Preprocessor is open source server-side scripting language that is widely used for web development. PHP scripts are executed on the server. PHP allows writing dynamically generated web pages efficiently and quickly. The syntax is mostly borrowed from C, Java and perl. PHP is free to download and use.
2	What is PEAR in php? BTL1 PEAR(PHP Extension and Application Repository) is a framework and repository for reusable PHP components. PEAR is a code repository containing all kinds of php code snippets and libraries. PEAR also offers a command-line interface that can be used to automatically install "packages".
3	Difference between \$message vs. \$\$message in PHP. BTL1 \$message is a variable with a fixed name. \$\$message is a variable whose name is stored in \$message. If \$message contains "var", \$\$message is the same as \$var.and call the document.form.submit() function in JavaScript code.
4	List the different types of errors in PHP. BTL1 Notices, Warnings and Fatal errors are the types of errors in PHP Notices: Notices represents non-critical errors, i.e. accessing a variable that has not yet been defined. By default, such errors are not displayed to the user at all but whenever required, you can change this default behavior. Warnings: Warnings are more serious errors but they do not result in script termination. i.e calling include() a file which does not exist. By default, these errors are displayed to the user. Fatal errors: Fatal errors are critical errors i.e. calling a non-existent function or class. These errors cause the immediate termination of the script. Explain the importance of the function htmlentities. The htmlentities() function converts characters to HTML entities.
5	What is MIME? BTL1 MIME - Multi-purpose Internet Mail Extensions. MIME types represents a standard way of classifying file types over Internet. Web servers and browsers have a list of MIME types, which facilitates files transfer of the same type in the same way, irrespective of operating system they are working in. A MIME type has two parts: a type and a subtype. They are separated by a slash (/). MIME type for Microsoft Word files is application and the subtype is msword, i.e. application/msword .
6	What is the difference between PHP and JavaScript? BTL1 The difference lies with the execution of the languages. PHP is server side scripting language, which means that it can't interact directly with the user. Whereas, JavaScript is client side scripting language, that is used to interact directly with the user..
7	What does ODBC do in context with PHP? BTL1 PHP supports many databases like dBase, Microsoft SQL Server, Oracle, etc. But, it also supports databases like filePro, FrontBase and InterBase with ODBC connectivity. ODBC stands for Open Database connectivity, which is a standard that allows user to

	communicate with other databases like Access and IBM DB2..
8	What are XML Parsers? [NOV/DEC 2017] [NOV/DEC 2015] BTL1 XML Parsers are used to check whether the document is well formed and valid.
9	How is XML parsing done with SAX? BTL1 A SAX parser is a mechanism for transforming an XML text document into stream of events corresponding to the markup and character data contained in the original document.
10	What are the types of XML Parsers? BTL There are two types : Validating and non validating parsers
11	What is well-formed document? BTL1 An HTML or XML document is said to be well formed when it contains elements with proper tagging and no syntactic errors.
12	What is DTD? [NOV/DEC 2019] BTL1 A Document Type Declaration enables an XML parser to verify whether an XML document is valid, i.e. its elements contain the proper attributes in the proper sequence.
13	What is XML Namespace? [NOV / DEC 2016] BTL1 An XML namespace is a collection of element and attribute names. Each namespace has a unique name that provides a means for document authors to unambiguously refer to elements with the same name in order to prevent collisions.
14	What is the use of XML declaration? BTL1 XML declaration is a special tag used to specify the version of XML used to write the document and optionally some additional meta-information about the document such as the character set/encoding used. For e.g the syntax of XML declaration is <? XML VERSION="1.0"?>
15	What is the purpose of XSLT? [MAY/JUN 2016] BTL1 • The XSLT stands for XSL Transformations and XSL stands for extensible Style sheet Language. The XSLT is used for defining the XML document transformation and presentations.
16	What are XML Schemas? BTL1 XML Schemas are part of the XML vocabulary and its addresses the standard for XML document validation by including a definition of a collection of standard data types which are used to describe data structures.
17	What Do You Mean by DTD IN XML? BTL1 • DTD means Document Type Definition.

	<ul style="list-style-type: none"> • DTD file is similar to CSS file, because DTD also contains only styles. DTD contains various styles which are to be applied in XML document. • Like .CSS file .DTD file also should be linked with XML program. Styles in XML program should be save with .xsl (Xml Style Sheet Language) extension.
18	<p>Define XML. BTL1</p> <p>XML is a meta- markup language that provides a format for describing structured data. This facilitates more structured declarations of content and more meaningful search results across multiple platforms.</p>
19	<p>When should the super global arrays in PHP be used? Which super global array in PHP would contain a HTML form's POST data? [MAY/JUN 2016] BTL1</p> <p>\$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods). PHP stores all global variables in an array called \$GLOBALS[index]. The <i>index</i> holds the name of the variable..</p>
PART B	
1	<p>Explain in detail about how to connect database with PHP? [NOV / DEC 2016] [13M]</p> <p>BTL 3</p> <p>Key Points:</p> <ol style="list-style-type: none"> 1.Creating a database[4M] 2. Selecting a database(2M) 3.Listing a database(3M) 4.Creating,altering and inserting into a table(4M) <p>Answer: Page No. 7-45 in Technical publication</p>
2	<p>Discuss the PHP variables and program control.[APRIL/MAY 2019] (13M)</p> <p>BTL 3</p> <p>Key Points:</p> <ol style="list-style-type: none"> 1. Variables[4M] 2. While,do while,for,selection statements(8M) <p>Answer: Page No. 7-4 in Technical publication</p>
3	<p>Discuss the concepts of XML Schema, built in and user defined data types in detail. [NOV / DEC 2016] [MAY/JUN 2016] [NOV/DEC 2017] [NOV/DEC 2015] BTL 3</p> <p>Key Points:</p> <ol style="list-style-type: none"> 3.Sample XML schema[6M] 4. Data types(2M) <p>Answer: Page No. 8-14 in Technical publication</p>
PART C	
1	<p>Discuss the XSL and XSLT Transformation. [NOV/DEC 2017]</p> <p>BTL 2</p> <p>Key Points:</p>

	<p>1.Transforming XML into XSLT[6M] 2. XSL elemnets (4M) 3.Displaying XML documents(4M)</p> <p>Answer: Page No. 8-33 in Technical publication</p>
2	<p>Explain the string comparison capability of PHP using regular expressions with an example. [NOV / DEC 2016] [NOV/DEC 2017] BTL 3</p> <p>Key Points:</p> <p>1.Preg_match and Preg_split example program[14M]</p> <p>Answer: Page No. 7-35 in Technical publication</p>

UNIT-5 INTRODUCTION TO AJAX and WEB SERVICES

AJAX: Ajax Client Server Architecture-XML Http Request Object-Call Back Methods;
 Web Services: Introduction- Java web services Basics → Creating, Publishing, Testing and Describing a Web services (WSDL)-Consuming a web service, Database Driven web service from an application –SOAP.

PART A

1	<p>What is a web service? Give any four examples. [APRIL/MAY 2019,2018]BTL1</p> <p>The web services are the software systems that are displayed by the web browser using the web protocol. These software systems are used by the some software applications rather than by tend users directly</p>
2	<p>What is SOAP? [APRIL/MAY 2018] BTL1</p> <p>Simple Object Access Protocol is simple XML based protocol which allows applications to exchange information over HTTP. Web services use this protocol</p>
3	<p>State advantages of AJAX BTL1</p> <p>AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.</p> <p>Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.</p>
4	<p>Define WSDL BTL1</p> <p>WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.</p>

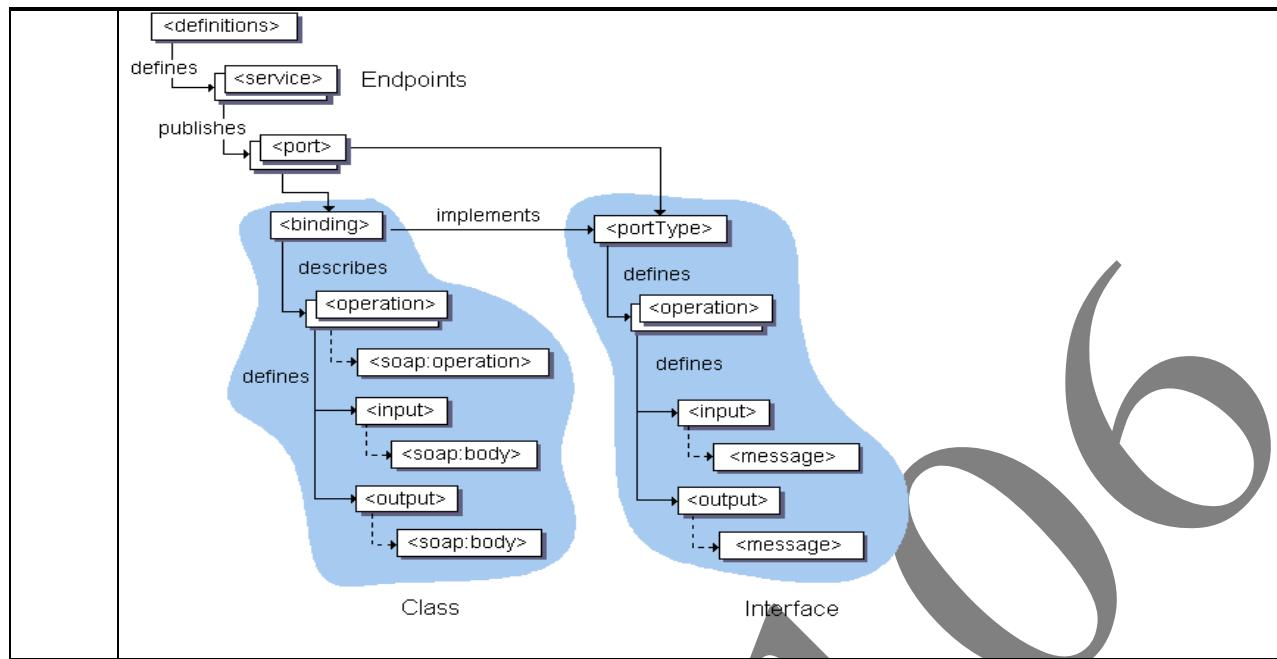
5	<p>Give Structure of SOAP message BTL1</p> <p>It consists of the following:</p> <ul style="list-style-type: none"> • The SOAP envelope element • The SOAP header element • The SOAP body element <p>The SOAP fault element</p>
6	<p>What is the role of callback function in performing a partial page update in an AJAX application? BTL1</p> <p>Partial page update are a part of callback functions which process returned data from a server and only update the parts of the page the callback says to update</p>
7	<p>Write Advantages of Ajax in web services BTL1</p> <ul style="list-style-type: none"> • Better interactivity • Easier navigation • Backed by reputed brands
8	<p>What is the use of XMLHttpRequestObject? BTL1</p> <p>It provides easy way to retrieve data from a URL without having to do a full page refresh</p>
9	<p>List some example of Webservices BTL1</p> <ul style="list-style-type: none"> • Weather forecast • Currency converter • Credit card validation <p>Resource Management system</p>
10	<p>What are all the controls of Ajax? BTL1</p> <ul style="list-style-type: none"> • Following are the controls of Ajax: • ScriptManager • ScriptManagerProxy • UpdatePanel • UpdateProgress
11	<p>What is the name of the DLL that contains Ajax control tool kit? [NOV/DEC 2018]</p> <p>BTL1</p> <p>Ajaxcontroltoolkit.dll is the DLL used for Ajax control tool kit and it can be downloaded from the internet. It can be added in the tool box or copied directly in the bin folder</p>
12	<p>Define - JAX-RPC BTL1</p>

	JAX-RPC [Java API for XML-Based RPC] is an application program interface [API] in the Java. Web Services Developer Pack [WSDP] that enables Java developers to include remote procedure calls.[RPCs] with Web services or other Web-based applications. JAXRPC is aimed at making it easier for applications or Web services to call other applications or Web services.
13	<p>Define - SOAP Fault element. BTL1</p> <p>The SOAP fault mechanism returns specific information about the error, including a predefined code, a description, the address of the SOAP processor that generated</p> <ul style="list-style-type: none"> • A SOAP Message can carry only one fault block • Fault element is an optional part of SOAP Message <p>SOAP fault is linked to the 500 to 599 range of status codes.</p>
14	<p>Define - SOAP Fault element. BTL1</p> <p>The SOAP fault mechanism returns specific information about the error, including a predefined code, a description, the address of the SOAP processor that generated</p> <ul style="list-style-type: none"> • A SOAP Message can carry only one fault block • Fault element is an optional part of SOAP Message <p>SOAP fault is linked to the 500 to 599 range of status codes.</p>
15	<p>Expand DOM, AJAX, AWT, JDBC. BTL1</p> <p>DOM-Document Object Model AJAX- Asynchronous Java and XML AWT-Abstract Window Tool Kit JDBC- Java Database Connectivity</p>
16	<p>List some web service technologies. BTL1</p> <ul style="list-style-type: none"> • Apache Axis • SOAP • JAX-RPC <p>.Net framework</p>
17	<p>List the port types of WSDL elements. BTL 1</p> <ul style="list-style-type: none"> • One-way • Request-response • Solicit-response <p>Notification</p>
18	<p>What do you mean by call back function? [APRIL/MAY 2019]BTL 1</p> <p>It is a function which is passed to another function as a parameter. The call back function</p>

	is called inside the other function.
19	What are the elements should be present in WSDL? [NOV/DEC 2018] BTL 1 Types <ul style="list-style-type: none"> • Messages • portType • Binding
20	What is binding in WSDL? BTL 1 Binding defines the name of the binding and type attributes specifies the port and specifies the name of class in the AWT event-class hierarchy.
21	Difference between SOAP and HTML. BTL 1 SOAP- used to build simple and useful API's called web services It uses XML information HTML is used for communication for the WWW and transfers data using internet It relies on structured text.
22	List the basic concept of JAX-RPC. BTL 1 The jax-rpc are the higher level technologies of web services. These are Java API for XML(JAX) based on remote procedure call.(RPC)
23	What is service endpoint interface in RPC? BTL 1 It's an interface that declared method that client can invoke while using the service
24	Define complex types. BTL 2 Complex types are objects created by the developer. Eg: customer with properties CustId and name is a complex type.
25	What is content type and content length? BTL 1 It is a headers of the HTTP protocol for HTTP GET and HTTP POST method to communicate.
PART B	
1	Explain Ajax client server architecture in detail with a diagram (13 M)(Nov/Dec'16,18) (APR/MAY 2019) BTL3 Answer: Page: 13-2 -A.A. Putembekar <ul style="list-style-type: none"> -AJAX - Asynchronous JavaScript and XML. -Group of JavaScript, DOM, XML, HTML, CSS etc. -Allows to send and receive data asynchronously without reloading the web page. Diagram <ol style="list-style-type: none"> 1. User sends a request from the UI - a javascript call goes to XMLHttpRequest object.

	<p>2. HTTP Request sent to the server by XMLHttpRequest object.</p> <p>3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.</p> <p>4. Retrieves Data</p> <p>5. Server sends XML data or JSON data to the XMLHttpRequest callback function. HTML and CSS data displayed on the browser.</p>
2	<p>Explain XMLHttpRequest Methods in detail (10M) (Nov/Dec'16) [APR/MAY 2019]</p> <p>BTL3</p> <p>Answer: Page:13-4 -A.A. Putembekar</p> <p>An object of XMLHttpRequest is used for asynchronous communication between client and server.(3M) It performs following operations: (3M)</p> <ul style="list-style-type: none"> 1. Sends data from the client in the background 2. Receives the data from the server 3. Updates the webpage without reloading it. <p>Methods (4M)</p>
3	<p>How to create, publish, and test web service in detail(13 m) (Nov/Dec'16) BTL3</p> <p>Answer: Page: 14-5 - A.A. Putembekar</p> <p>Writing webservice (3M)</p> <p>Writing web service client (3M)</p> <p>A WSDL document is used to describe a web service. This description is required, so that client applications are able to understand what the web service actually does .(4M)</p> <ul style="list-style-type: none"> • The WSDL file contains the location of the web service and The methods which are exposed by the web service. <p>Below is the general structure of a WSDL file (3 M)</p> <ul style="list-style-type: none"> • Definition • TargetNamespace • DataTypes • Messages • Porttype • Bindings <p>Service</p>
4	<p>Define SOAP and explain the building blocks of SOAP(13 M) (Nov/Dec'15,17,18)</p> <p>BTL4</p>

	<p>Answer: Page:14-24 - A.A. Putembekar</p> <p>SOAP is an acronym for Simple Object Access Protocol. It is an XML-based messaging protocol for exchanging information among computers. SOAP is an application of the XML specification. (4 M)</p> <p>SOAP</p> <table> <tr> <td>Structure of SOAP</td><td>(4 M)</td></tr> <tr> <td>SOAP AND HTTP</td><td>(2 M)</td></tr> <tr> <td>SOA&D ENCODING</td><td>(1 M)</td></tr> <tr> <td>RPC REPRESENTATION</td><td>(2M)</td></tr> </table>	Structure of SOAP	(4 M)	SOAP AND HTTP	(2 M)	SOA&D ENCODING	(1 M)	RPC REPRESENTATION	(2M)
Structure of SOAP	(4 M)								
SOAP AND HTTP	(2 M)								
SOA&D ENCODING	(1 M)								
RPC REPRESENTATION	(2M)								
	<p>PART C</p>								
1	<p>Explain java web services in detail (15 M) BTL3</p> <p>Answer: Page: 14-3 - A.A. Putembekar</p> <p>Java Web Services (3M)</p> <p>Java provides - own API to create both SOAP as well as REST web services.</p> <p>JAX-WS: JAX-WS stands for Java API for XML Web Services (3 M)</p> <p>JAX-RS: Java API for RESTful Web Services (JAX-RS) - Java API for creating REST web services. JAX-RS uses annotations to simplify the development and deployment of web services. (3 M)</p> <p>PROGRAM (6 M)</p>								
2	<p>Give Structure of WSDL (15m)</p> <p>M)(Apr/May'18)(Nov/Dec'16)BTL3 Answer: Page:</p> <p>14-2 - A.A. Putembekar</p>								



JIT - 2106

OBJECTIVES:

- To understand the various characteristics of Intelligent agents
- To learn the different search strategies in AI
- To learn to represent knowledge in solving AI problems
- To understand the different ways of designing software agents
- To know about the various applications of AI.

UNIT I	INTRODUCTION	9
Introduction–Definition - Future of Artificial Intelligence – Characteristics of Intelligent Agents– Typical Intelligent Agents – Problem Solving Approach to Typical AI problems.		
UNIT II	PROBLEM SOLVING METHODS	9
Problem solving Methods - Search Strategies- Uninformed - Informed - Heuristics - Local Search Algorithms and Optimization Problems - Searching with Partial Observations - Constraint Satisfaction Problems – Constraint Propagation - Backtracking Search - Game Playing - Optimal Decisions in Games – Alpha - Beta Pruning - Stochastic Games		
UNIT III	KNOWLEDGE REPRESENTATION	9
First Order Predicate Logic – Prolog Programming – Unification – Forward Chaining- Backward Chaining – Resolution – Knowledge Representation - Ontological Engineering- Categories and Objects – Events - Mental Events and Mental Objects - Reasoning Systems for Categories - Reasoning with Default Information		
UNIT IV	SOFTWARE AGENTS	9
Architecture for Intelligent Agents – Agent communication – Negotiation and Bargaining – Argumentation among Agents – Trust and Reputation in Multi-agent systems.		
UNIT V	APPLICATIONS	9
AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing - Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving.		

TOTAL :45

PERIODS OUTCOMES:

- Upon completion of the course, the students will be able to:
- Use appropriate search algorithms for any AI problem
- Represent a problem using first order and predicate logic
- Provide the apt agent strategy to solve a given problem
- Design software agents to solve a problem
- Design applications for NLP that use Artificial Intelligence.

TEXT BOOKS:

- 1 S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice Hall, Third Edition, 2009.
- 2 I. Bratko, —Prolog: Programming for Artificial Intelligence, Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.

REFERENCES:

1. M. Tim Jones, —Artificial Intelligence: A Systems Approach (Computer Science), Jones and Bartlett Publishers, Inc.; First Edition, 2008
2. Nils J. Nilsson, —The Quest for Artificial Intelligence, Cambridge University Press, 2009.

SubjectCode:CS8691
Subject Name: ARTIFICIAL INTELLIGENCE

Year/Semester: III/06
Subject Handler: Ms.N.R GLADISS MERLIN

UNIT -1- INTRODUCTION	
Introduction–Definition - Future of Artificial Intelligence – Characteristics of Intelligent Agents – Typical Intelligent Agents – Problem Solving Approach to Typical AI problems.	
PART * A	QUESTIONS
1.	<p>What is artificial intelligence? BTL1 The exciting new effort to make computers think machines with minds in the full and literal sense. Artificial intelligence systemizes and automates intellectual tasks and is therefore potentially relevant to any sphere of human intellectual activities.</p>
2.	<p>Define Turing test. BTL1 The Turing test proposed by Alan Turing was designed to provide a satisfactory operational definition of intelligence. Turing defined intelligent behavior as the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator.</p>
3.	<p>List the capabilities that a computer should possess for conducting a Turing Test. BTL1 The capabilities that a computer should possess for conducting a Turing Test are,</p> <ul style="list-style-type: none"> ✓ Natural Language Processing; ✓ Knowledge Representation; ✓ Automated Reasoning; ✓ Machine Language.
4.	<p>Define an agent. BTL1 An agent is anything that can be viewed as perceiving its environment through sensors and acting upon the environment through effectors.</p>
5.	<p>Define rational agent. (DEC 2011) (APRIL/MAY 2015) BTL1 A rational agent is one that does the right thing. Here right thing is one that will cause agent to be more successful. That leaves us with the problem of deciding how and when to evaluate the agent's success.</p>
6.	<p>Define an Omnipotent agent. BTL1 An omnipotent agent knows the actual outcome of its action and can act accordingly; but omniscience is impossible in reality.</p>
7.	<p>What are the factors that a rational agent should depend on at any given time? BTL2 The factors that a rational agent should depend on at any given time are,</p> <ul style="list-style-type: none"> ✓ The performance measure that defines criterion of success; ✓ Agent's prior knowledge of the environment; ✓ Action that the agent can perform; ✓ The agent's percept sequence to date.
8.	<p>List the measures to determine agent's behavior. BTL1 The measures to determine agent's behavior are,</p> <ul style="list-style-type: none"> ✓ Performance measure, ✓ Rationality, ✓ Omnipotence

9.	<p>Recognise the various types of agent programs. (DEC 2012) BTL1</p> <p>The various types of agent programs are,</p> <ul style="list-style-type: none"> ✓ Simple reflex agent program; ✓ Agent that keep track of the world; ✓ Goal based agent program; <p>Utility based agent program.</p>
10.	<p>Name the components of a learning agent.BTL1</p> <p>The components of a learning agent are,</p> <ul style="list-style-type: none"> ✓ Learning element; ✓ Performance element; ✓ Critic; <p>Problem generator.</p>
11.	<p>List out some of the applications of Artificial Intelligence. BTL1</p> <p>Some of the applications of Artificial Intelligence are,</p> <ul style="list-style-type: none"> ✓ Autonomous planning and scheduling; ✓ Game playing; ✓ Autonomous control; ✓ Diagnosis; ✓ Logistics planning; <p>Robotics.</p>
12.	<p>What is depth-limited search? BTL1</p> <p>Depth-limited avoids the pitfalls of DFS by imposing a cut off of the maximum depth of a path. This cutoff can be implemented by special depth limited search algorithm or by using the general search algorithm with operators that keep track of the depth.</p>
13.	<p>Define breadth-first search. BTL1</p> <p>The breadth-first search strategy is a simple strategy in which the root-node is expanded first, and then all the successors of the root node are expanded, then their successors and so on. It is implemented using TREE-SEARCH with an empty fringe that is a FIFO queue, assuring that the nodes that are visited first will be expanded first.</p>
14.	<p>Describe problem formulation. BTL1</p> <p>Problem formulation is the process of deciding what actions and states to consider for a goal that has been developed in the first step of problem solving.</p>
15.	<p>List the four components of a problem.BTL1</p> <p>The four components of a problem are,</p> <ul style="list-style-type: none"> ✓ An initial state; ✓ Actions; □ Goal test
16.	<p>Define iterative deepening search. BTL1</p> <p>Iterative deepening is a strategy that sidesteps the issue of choosing the best depth limit by trying all possible depth limits: first depth 0, then depth 1, then depth 2& so on.</p>
17.	<p>Mention the criteria's for the evaluation of search strategy.(MAY/JUNE 2014) BTL2</p> <p>The criteria's for the evaluation of search strategy are,</p> <ul style="list-style-type: none"> ✓ Completes; ✓ Time; ✓ complexity; ✓ Space complexity;

	Optimality.
18.	<p>Define the term percept. BTL1 The term percept refers to the agents perceptual inputs at any given instant. An agent's percept sequence is the complete history of everything that the agent has perceived.</p>
19.	<p>What is Constraint Satisfaction Problem. BTL2 A constraint satisfaction problem is a special kind of problem satisfies some additional structural properties beyond the basic requirements for problem in general. In a CSP, the states are defined by the values of a set of variables and the goal test specifies a set of constraint that the value must obey.</p>
20.	<p>List some of the uninformed search techniques. BTL1 Some of the uninformed search techniques are,</p> <ul style="list-style-type: none"> ✓ Breadth-First Search(BFS); ✓ Depth-First Search(DFS); ✓ Uniform Cost Search; ✓ Depth Limited Search; ✓ Iterative Deepening Search; <p>Bidirectional Search.</p>
21.	<p>Define Abstraction. (May 2012) BTL1 Abstraction is the process by which data and programs are defined with a representation similar in form to its meaning (semantics), while hiding away the implementation details. Abstraction tries to reduce and factor out details so that the programmer can focus on a few concepts at a time. A system can have several abstraction layers whereby different meanings and amounts of detail are exposed to the programmer. For example, low-level abstraction layers expose details of the computer hardware where the program runs, while high-level layers deal with the business logic of the program.</p>
22.	<p>What does Software Agent mean? (NOV/DEC 2013) BTL1 A software agent is a piece of software that functions as an agent for a user or another program, working autonomously and continuously in a particular environment. It is inhibited by other processes and agents, but is also able to learn from its experience in functioning in an environment over a long period of time.</p>
23.	<p>Define the effect of heuristic accuracy on performance. (NOV/DEC 2013) BTL1 A heuristic is a method that might not always find the best solution but is guaranteed to find a good solution in reasonable time. By sacrificing completeness it increases efficiency. Useful in solving tough problems which could not be solved any other way. Solutions take an infinite time or very long time to compute. The classic example of heuristic search methods is the travelling salesman problem.</p>
	PART * B
1	<p>Explain in detail any of the four agent structure (or) Describe in detail about the following:</p> <ol style="list-style-type: none"> i) Simple reflex agent.(3M) ii) Model based agent.(3M) iii) Utility based agent.(3M) iv) Goal based agent(4M)(Dec 2012) BTL1 <p>Answer:Page. 18-Elaine Rich</p> <p>The various types of agent programs are,</p> <ul style="list-style-type: none"> ✓ Simple reflex agent program; (1M) ✓ Diagram (1M) <p>function SIMPLE-REFLEX-AGENT(percept)</p>

	<p>returns an action Static: rules, a set of condition – action rules</p> <pre> State <- INTERPRET – INPUT (Percept) Rule <- RULE-MATCH (STATE, RULES) Action <- RULE- ACTION[rule] return action ✓ Model based reflex agent;(2M) ✓ Diagrams (2M) ✓ Goal based agent program; (2M) ✓ Diagram (2M) ✓ Utility based agent program.(2M) ✓ Diagram (1M) </pre>
2	<p>Explain in detail Model based reflex agent.(13M)(May 2012)BTL2</p> <p>Answer:Page. 18-Elaine Rich</p> <p>Diagram</p> <p>(3M)</p>
	<pre> function REFLEX-AGENT-WITH-STATE(percept) returns an action static: state, a description of the current world state rules, a set of condition-action rules action, the most recent action, initially none state ← UPDATE-STATE(state, action, percept) rule ← RULE-MATCH(state, rules) action ← RULE-ACTION[rule] return action </pre> <p>Explanation - it keeps track of the current state of the world-using an internal model- then it chooses and action – as reflex agent (10M)</p>
3	<p>Inference the Goal based reflex agent.(13M)(May 2012)BTL2</p> <p>Answer:Page. 18-Elaine Rich</p> <p>Diagram</p> <p>Algorithm</p> <p>Explanation -it keeps track of the world state as well as a set of goals it is trying to achieve and chooses an action that will eventually lead to the achievement of the goal. (5M)</p>
4	<p>Explain in detail learning agent.(13M)(May 2013)BTL2</p> <p>Answer:Page. 18-Elaine Rich</p> <p>Diagram (5M)</p> <p>Explanation</p> <p>A learning agent can be divided into four conceptual components</p> <ul style="list-style-type: none"> ✓ Element responsible for making improvements(2M) ✓ performance element, which is responsible for selecting external actions. (2M) ✓ The learning element (2M) <p>CRITIC feedback from the critic on how the agent is doing and determines how the performance element</p>

	should be modified to do better in the future. (2M)
5	<p>Summarize depth-First search .(13M)(May 2012)BTL2</p> <p>Answer:Page. 160-Elaine Rich</p> <ul style="list-style-type: none"> ✓ Explanation - extends the current path as far as possible before backtracking(5M) ✓ Algorithm (5M) ✓ Working – example diagram (3M)
6	<p>Define CSP and Discuss about backtracking search for CSPs.(13M)(APRIL/MAY 2015)BTL2</p> <p>Answer: Page. 68-Elaine Rich</p> <p>Types explanation -A Constraint Satisfaction Problem(or CSP) is defined by a set of variables ,X_1, X_2, \dots, X_n,and a set of constraints C_1, C_2, \dots, C_m. Each variable X_i has a nonempty domain D_i of possible values.</p> <p>Each constraint C_i involves some subset of variables and specifies the allowable combinations of values for that subset(5M)</p> <p>Working flow(3M)</p> <p>Explanation- CSP can be viewed as a standard search problem as follows :</p> <ul style="list-style-type: none"> ✓ Initial state : the empty assignment {},in which all variables are unassigned. Successor function : a value can be assigned to any unassigned variable, provided that it does not conflict with previously assigned variables.
PART * C	
	<ul style="list-style-type: none"> ✓ How did you describe PEAS description for at least four agent types? (7M) BTL1 ✓ How did you describe PEAS? (8M) BTL1 <p>Answer:Page.4-Elaine Rich</p> <p>PEAS description(3M)</p> <p>Performance Measure – measuring the performances (3M)</p> <p>Environment – the environment for the agent to act in (3M)</p> <p>Actuators – the agent act through the actuators (3M)</p> <p>Sensors – the agent perceives through the sensor (3M)</p>
	<p>Summarize in detail about production system characteristics. (15M) BTL2</p> <p>Answer : Page : Page.6-Elaine Rich</p> <p>Explanation –</p> <ul style="list-style-type: none"> ✓ It is the good way to describe the operation(3M) ✓ Monotonic production system(2M) ✓ Non monotonic production system(2M) ✓ Partially commutative system(2M) ✓ Commutative production system(2M) ✓ Non monotonic partially commutative system(2M) ✓ Production systems not partially commutative(2M)
UNIT II PROBLEM SOLVING METHODS	
	Problem solving Methods - Search Strategies- Uninformed - Informed - Heuristics - Local Search Algorithms and Optimization Problems - Searching with Partial Observations - Constraint Satisfaction Problems – Constraint Propagation - Backtracking Search - Game Playing - Optimal Decisions in Games – Alpha - Beta Pruning - Stochastic Games

1	<p>What are the standard quantifiers of First Order Logic? BTL1 The First Order Logic contains two standard quantifiers.</p> <p>They are: Universal Quantifiers \forall Existential Quantifiers \exists</p>
2	<p>Define Universal Quantifier with an example. BTL1 To represent —All elephants are mammal—Raj is an elephant—is represented by Elephant(Raj) and —Raj is a mammal. The first order logic is given by X Elephant(x) => Mammal(x) Refers to —For all. P is any logical expression, which is equivalent to the conjunction(i.e. the) of all sentences obtained by substituting the name of an object for the variable x where it appears in p. The above sentence is equivalent to Elephant(Raj)=>Mammal(Raj) Elephant(John)=>Mammal(John) Thus it is true if and only if, all the above sentences are true that is if p is true forall objects x in the universe. Hence, is called universal quantifier.</p>
3	<p>Recognize is the use of equality symbol? BTL1 The equality symbol is used to make the statements more effective that two terms refer to the same object. Eg: Father (John)=Henry</p>
4	<p>Define Higher Order Logic. BTL1 The Higher Order Logic allows quantifying over relations and functions as well as over objects. Eg: The two objects are equal if and only if, all the properties to them are equivalent.x, y(x=y)(p p(x) p(y))</p>
5	<p>Describe the first Order Logic. BTL1 First Order Logic, a representation language that is far more powerful than propositional logic. First Order Logic commits to the existence of objects and relations. Eg: One plus two equals three Objects – one,two& three Relations-equals Functions-plus</p>
6	<p>What is called declarative approach? BTL1 The representation language makes it easy to express the knowledge in the form of sentences. This simplifies the construction problem enormously. This is called as declarative approach.</p>
7	<p>State the aspects of a knowledge representation language. BTL2 A knowledge representation language is defined in two aspects: Syntax: The syntax of a language describes the possible configuration that canconstitute sentences. Semantics: It determines the facts in the world to which the sentences refer.</p>
8	<p>What is called entailment? BTL2 The generations of new sentences that are necessarily true given the old sentences are true. This relation between sentences is called entailment.</p>
9	<p>What is meant by tuple? BTL1 A tuple is a collection of objects arranged in a fixed order and is written with angle brackets surrounding the objects. {<Richard the Lionheart, King John>, <King John, Richard the Lion heart>}</p>
10	<p>Define propositional Logic? BTL1 Propositional Logic is a declarative language because its semantics is based on a truth relation between sentences and possible worlds. It also has sufficient expressive power to deal with partial information, using disjunction and negation.</p>
11	<p>What is compositionality in propositional logic? BTL2 Propositional Logic has a third property that is desirable in representation languages, namely compositionality. In a compositionality language, the meaning ofsentences is a function of the</p>

	meaning of its parts. For example, —S1^S2 is related to the meanings of —S1 and S2 .
12	Define Symbols. BTL1 The basic syntactic elements of first order logic are the symbols that stand for objects, relations and functions. The symbols are in three kinds. Constant symbols which stand for objects, Predicate symbols which stand for relations and Function symbol which stand for functions.
13	Describe ground term, Inference. BTL1 The term without variables is called ground term. The task of deriving the new sentence from the old is called Inference.
14	Describe Data log. BTL1 The applications of query tools are The set of first order definite clauses with no function symbols is called datalog. Eg: -The country Nono, an enemy of America Enemy(Nono, America) The absence of function symbols makes inference much easier.
15	What is Pattern Matching? BTL2 The —inner loop of the algorithm involves finding all possible unifiers such that the premise of a rule unifies with a suitable set of facts in the knowledge base. This is called Pattern Matching.
	PART * B
1	Give the Syntax and Semantics of a first order logic in detail with an eg. Explain predicate logic (May 2013) (13M) BTL3 Answer: Page. 82-Elaine Rich <ul style="list-style-type: none"> ✓ Introduction- The best way to find usage of First order logic is through examples. The examples can be taken from some simple domains. In knowledge representation, a domain is just some part of the world about which we wish to express some knowledge. (3M) ✓ Assertions and queries in first-order logic (5M) ✓ Example (5M)
2	Discuss about the following: Greedy best-first search. (4M) A* search (4M) Memory bounded heuristic search. (5M) BTL2 Answer: Page. 166-Elaine Rich Greedy best-first search. <ul style="list-style-type: none"> ✓ Explanation (2M) ✓ Working (2M) ✓ A* search Explanation(2M) ✓ Working (2M) ✓ Explanation (2M) ✓ Working (3M)
3	Interpret the uninformed search strategies. (13M)(May/June2009) BTL2 Answer: Page. 57-Elaine Rich Introduction (3M) <ul style="list-style-type: none"> ✓ No additional information ✓ beyond that provided in the ✓ problem definition ✓ Not effective ✓ No information about number of steps or path cost <ul style="list-style-type: none"> ✓ Breath first search (2M) ✓ Depth first search (2M) ✓ Depth limited search (2M) ✓ Iterative deepening search (2M)

	<p>✓ Bi directional search (2M)</p>
PART * C	
1	<p>Explain crypt arithmetic problem for the below Problem: SEND +MORE</p> <p>.....</p> <p>Initial state: MONEY</p> <p>No two letters have the same value. The sums of the digits must be shown in the problem. (15M) BTL4</p> <p>Answer: Page. Notes</p> <p>S=9 E=5 N=6 D=7 M=1 O=0 R=8 Y=2</p>
2	<p>Explain crypt arithmetic problem for the below Problem: CROSS +DANGER</p> <p>.....</p> <p>Initial state: CROSS</p> <p>No two letters have the same value. The sums of the digits must be shown in the problem. (15M) BTL4</p> <p>Answer:Page. Notes</p> <p>C=9 R=6 O=2 S=3 R=6 A=5 D=1 N=8 G=7 E=4</p>
3	<p>i) Solve any two Informed Search Strategies with an example. (7M) (May/June 2016) BTL3</p> <p>ii) Explain the algorithm for generate and test simple hill climbing. (8M) (May/June 2016) BTL2</p> <p>Answer:Page. 52-Elaine Rich</p> <p>Informed Search Strategies</p> <ul style="list-style-type: none"> ✓ Heuristic search(3M) ✓ Generate and test(3M) ✓ Hill climbing(2M) ✓ Best first search(2M) ✓ Problem reduction(3M) ✓ Constraint satisfaction(2M)
	UNIT III KNOWLEDGE REPRESENTATION
	<p>First Order Predicate Logic – Prolog Programming – Unification – Forward Chaining- Backward Chaining – Resolution – Knowledge Representation - Ontological Engineering- Categories and Objects – Events - Mental Events and Mental Objects - Reasoning Systems for Categories - Reasoning with Default Information</p>
1.	<p>What are representations types of knowledge? BTL2</p> <p>Four General Representation Types</p> <ul style="list-style-type: none"> ✓ Logical Representations ✓ Semantic Networks ✓ Production Rules <p>Frames</p>
2.	<p>Describe a frame ? BTL1</p> <p>Frames represent an alternative way to structure and organise knowledge. A frame system is a hierarchy of frames. Each Frame has:</p> <ul style="list-style-type: none"> ✓ a name <p>slots: properties of the entity that has the name, and their values.</p>

3.	Define Uncertainty. BTL1 Uncertainty means that many of the simplifications that are possible with deductive inference are no longer valid.
4.	State the reason why first order, logic fails to cope with that the mind like Medical diagnosis. BTL3 Three reasons: <ul style="list-style-type: none">✓ Laziness: It is hard to lift complete set of antecedents of consequence, needed to ensure and exception less rule.✓ Theoretical Ignorance: Medical science has no complete theory for the domain. Practical ignorance: Even if we know all the rules, we may be uncertain about a particular item needed.
5.	Demonstrate is the need for probability theory in uncertainty? BTL3 Probability provides the way of summarizing the uncertainty that comes from our laziness and ignorance. Probability statements do not have quite the same kind of semantics known as evidences.
6.	Demonstrate is the need for utility theory in uncertainty? BTL1 Utility theory says that every state has a degree of usefulness, or utility to In agent, and that the agent will prefer states with higher utility. The use utility theory to represent and reason with preferences.
7.	What Is Called As Decision Theory? BTL2 Preferences As Expressed by Utilities Are Combined with Probabilities in the General Theory of Rational Decisions Called Decision Theory. Decision Theory = Probability Theory + Utility Theory.
8.	Define conditional probability. BTL1 Once the agents has obtained some evidence concerning the previously unknown propositions making up the domain conditional or posterior probabilities with the notation $p(A/B)$ is used. This is important that $p(A/B)$ can only be used when all be is known.
9.	When probability distribution is used? BTL3 If we want to have probabilities of all the possible values of a random variable probability distribution is used. Eg: $P(\text{weather}) = (0.7, 0.2, 0.08, 0.02)$. This type of notations simplifies many equations.
10.	What is an atomic event? BTL2 An atomic event is an assignment of particular values to all variables, in other words, the complete specifications of the state of domain.
11.	Define joint probability distribution. BTL1 Joint probability distribution completely specifies an agent's probability assignments to all propositions in the domain. The joint probability distribution $p(x_1, x_2, \dots, x_n)$ assigns probabilities to all possible atomic events; where x_1, x_2, \dots, x_n = variables.
12.	What is meant by belief network? BTL3 <ul style="list-style-type: none">✓ A belief network is a graph in which the following holds✓ A set of random variables✓ A set of directive links or arrows connects pairs of nodes.✓ The conditional probability table for each node The graph has no directed cycles.
13.	What are called as Poly trees? BTL2 The algorithm that works only on singly connected networks known Poly trees. Here at most one undirected path between any two nodes is present.
14.	What is a multiple connected graph? BTL2 A multiple connected graph is one in which two nodes are connected by more than one path.

15.	List the three basic classes of algorithms for evaluating multiply connected graphs. BTL1 The three basic classes of algorithms for evaluating multiply connected graphs <ul style="list-style-type: none"> ✓ Clustering methods; ✓ Conditioning methods; Stochastic simulation methods.
16.	What is called as principle of Maximum Expected Utility (MEU)? BTL2 The basic idea is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over all the possible outcomes of the action. This is known as MEU
17.	What is meant by deterministic nodes? BTL2 A deterministic node has its value specified exactly by the values of its parents, with no uncertainty.
18.	What are all the uses of a belief network? BTL3 The uses of a belief network are, <ul style="list-style-type: none"> ✓ Making decisions based on probabilities in the network and on the agent's utilities; ✓ Deciding which additional evidence variables should be observed in order to gain useful information; ✓ Performing sensitivity analysis to understand which aspects of the model have the greatest impact on the probabilities of the query variables (and therefore must be accurate); Explaining the results of probabilistic inference to the user.
19.	What is called as Markov Decision problem? BTL2 The problem of calculating an optimal policy in an accessible, stochastic environment with a known transition model is called a Markov Decision Problem (MDP).
20.	Define Dynamic Belief Network. BTL2 A Belief network with one node for each state and sensor variable for each time step is called a Dynamic Belief Network.(DBN).
21.	Define Dynamic Decision Network.BTL1 A decision network is obtained by adding utility nodes, decision nodes for action in DBN. DDN calculates the expected utility of each decision sequence.
22.	What is Bayesian Networks? (MAY/JUNE2016) BTL1 Graphical model for reasoning under uncertainty ○ Nodes represents variables ○ Arc represents direct connections between variables.
23.	Write the properties of fuzzy sets. (MAY/JUNE2016) BTL2 Properties: <ul style="list-style-type: none"> ✓ Distributivity Associativity ✓ Commutativity Idempotency
24.	What are representations types of knowledge? BTL1 Four General Representation Types <ul style="list-style-type: none"> ✓ Logical Representations ✓ Semantic Networks ✓ Production Rules Frames
25.	List the three basic classes of algorithms for evaluating multiply connected graphs.BTL1 The three basic classes of algorithms for evaluating multiply connected graphs <ul style="list-style-type: none"> ✓ Clustering methods; ✓ Conditioning methods; Stochastic simulation methods.
	PART * B
	(i) Summarize about Minimax algorithm in detail.(6M)

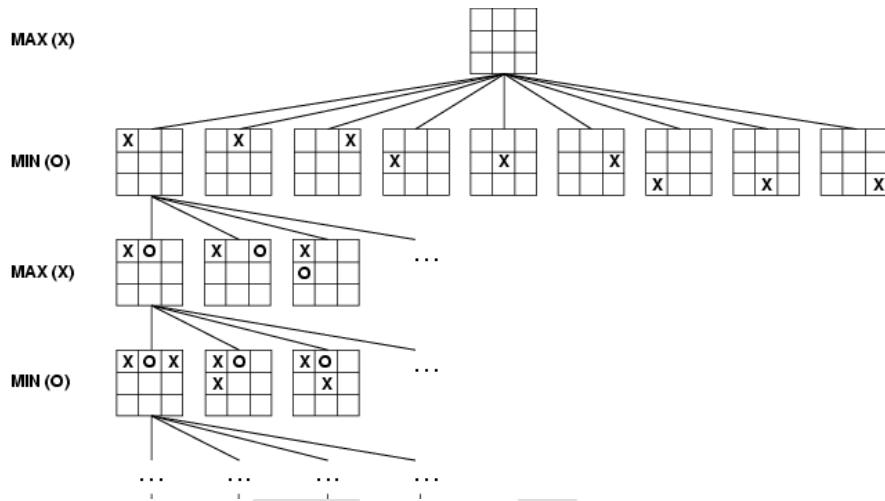
BTL2

(ii) Discuss Alpha, Beta pruning and Alpha-Beta algorithm.(7M) BTL2

Answer:Page. 233,236-Elaine Rich

- ✓ Minimax algorithm

consider games with two players, whom we will call MAX and MIN. MAX moves first, and then they take turns moving until the game is over. At the end of the game, points are awarded to the winning player and penalties are given to the loser. (6M)



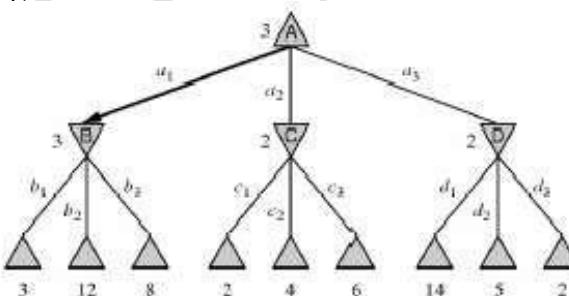
- ✓ Alpha, Beta pruning

- ✓ Introduction-
MAX

AMX_s turn to
utility values f
the other node

MIN

- ✓ Diagram (5M)



ich it is
des show the

ve at the root is
MIN_s best reply
M)

1 **Describe in detail the steps involved in the knowledge engineering process.**

(Apr/May11)(13M) BTL2

Answer:Page. 79-Elaine Rich

A knowledge engineer is someone who investigates a particular domain, learns what concepts are important in that domain, and creates a formal representation of the objects and relations in the domain. We will illustrate the knowledge engineering process in an electronic circuit domain that should already be fairly familiar. (3M)

Steps(10M)

- ✓ Identify the task.
- ✓ Assemble the relevant knowledge.
- ✓ Decide on a vocabulary of predicates, functions, and constants
- ✓ Encode general knowledge about the domain.
- ✓ Encode a description of the specific problem instance
- ✓ Pose queries to the inference procedure and get answers.
- ✓ Debug the knowledge base.

2	<p>Explain unification algorithm used for reasoning under predicate logic with an example. (Apr/May11)(13M) BTL2</p> <p>Answer:Page. 99,108-Elaine Rich</p> <p>The use of unification to identify appropriate substitutions for variables eliminates the instantiation step in first-order proofs, making the process much more efficient.(1M)</p> <ul style="list-style-type: none"> ✓ Completeness theorem (2M) ✓ Incompleteness (2M) ✓ Steps (8M) ✓ Move forward <ul style="list-style-type: none"> ¬ $\exists x p$ becomes for all $x \neg p$ ¬ $\exists x p$ becomes for all $x \neg p$ ✓ Standardize variables ✓ Elimination of extensional qualifiers ✓ Drop universal qualifiers <p>Learning heuristics</p>
3	<p>Demonstrate forward chaining and backward chaining in detail for a first order definite Clauses. (May 2012) (13M)BTL3</p> <p>Answer:Page. 134-Elaine Rich</p> <p>Introduction –</p> <ul style="list-style-type: none"> ✓ Using a deduction to reach a conclusion from a set of antecedents is called forward chaining. (2M) ✓ Example (6M) <p>Function FOL-FC-ASK (KB, α) returns a substitution or false Inputs: KB, the knowledge base, a set of first – order definite clauses A, the query, an atomic sentence</p> <p>Local variables: new, the new sentences inferred on each iteration Repeat until new is</p> <p>For each sentence r in KB do – APART(r)</p> <p>For each θ such that SUBSET ($\theta, P_1 \wedge \dots \wedge P_n$) = SUBSET($\theta, P_1 \wedge \dots \wedge P_1$)</p> <p>If q_1 is not a renaming of some sentence already in KB or new then do Add q_1 to new</p> <p>If ϕ is not fail then</p> <p>return ϕ Add new to</p> <p>KB Return false</p> <p>Algorithm (5M)</p>
4	<p>Calculate the completeness proof of resolution. (Nov/Dec 2014).(13M)BTL3</p> <p>Answer:Page. 108-Elaine Rich</p> <p>The generalized resolution inference rule provides a complete proof system for first order logic, using knowledge bases in conjunctive normal form.(1M)</p> <ul style="list-style-type: none"> ✓ Completeness theorem (2M) ✓ Incompleteness theorem(2M) Steps(8M) ✓ Move forward <ul style="list-style-type: none"> ¬ $\exists x p$ becomes for all $x \neg p$ ¬ $\exists x p$ becomes for all $x \neg p$ ✓ Standardize variables ✓ Elimination of extensional qualifiers ✓ Drop universal qualifiers <p>Learning heuristics</p>

	PART * C
1	<p>Explain resolution in predicate logic with suitable example.(15M) BTL5</p> <p>Answer:Page. 108-Elaine Rich</p> <p>The use of unification to identify appropriate substitutions for variables eliminates the instantiation step in first-order proofs, making the process much more efficient. (3M)</p> <ul style="list-style-type: none"> ✓ Completeness theorem (2M) ✓ Incompleteness theorem (2M) ✓ Steps (8M) ✓ Move forward <ul style="list-style-type: none"> ¬ $\exists x p \text{ becomes for all } x \neg p$ ¬ $\exists x p \text{ becomes for all } x \neg p$ ✓ Standardize variables ✓ Elimination of extensional qualifiers <p style="text-align: right;">Drop universal qualifiers</p>
2	<p>Consider the following sentences:</p> <ul style="list-style-type: none"> ✓ John like all kinds of food ✓ Apples are food ✓ Chicken is food ✓ Anything anyone eats and isn't killed is food ✓ Bill eats peanuts and still alive ✓ Sue eats everything Bill eats <p>i) Translate these sentences into formulae in predicate logic.</p> <p>ii) Convert the above FOL into clause form. (15M) BTL6</p> <p>Answer:Notes</p> <ul style="list-style-type: none"> ✓ Translate these sentences into formulae in predicate logic.(8M) ✓ Convert the above FOL into clause form(7M)
3	<p>Develop the following well-formed clause form with sequence of steps.</p> <p>$\forall x: [\text{Roman}(x) \wedge \text{know}(x, \text{Marcus})]$ $\rightarrow [\text{hate}(x, \text{Caesar}) \vee (\forall y: \exists z: \text{hate}(y, z))]$ $\rightarrow [\text{thinkcrazy}(x, y)]$ BTL6</p> <p>Answer: Notes</p> <ul style="list-style-type: none"> ✓ Translate these sentences into formulae in predicate logic.(8M) ✓ Convert the above FOL into clause form
4	<p>Analyze on resolution principles and procedures. (15M) BTL4</p> <p>Answer:Page. 108-Elaine Rich</p> <p>The use of unification to identify appropriate substitutions for variables eliminates the instantiation step in first-order proofs, making the process much more efficient. (3M)</p> <ul style="list-style-type: none"> ✓ Completeness theorem (2M) ✓ Incompleteness theorem (2M) ✓ Steps (8M) ✓ Move forward <ul style="list-style-type: none"> ¬ $\exists x p \text{ becomes for all } x \neg p$ ¬ $\exists x p \text{ becomes for all } x \neg p$ ✓ Standardize variables ✓ Elimination of extensional qualifiers

	<ul style="list-style-type: none"> ✓ Drop universal qualifiers <p>Learning heuristics</p>
5	<p>What is the entire temporal model. In temporal model explain filtering and prediction (Dec 2012) (May 2012)(13 M)BTL3</p> <p>Answer: Page. 188-Elaine Rich</p> <p>A learning algorithm is good if it produces hypothesis that do a good job of predicting the classification of example prediction quality can be estimated in advance on it can be estimated (3M)</p> <ul style="list-style-type: none"> ✓ Collect a large set of example. (2M) ✓ Divide it into two disjoint sets, the training set and the testset. (2M) ✓ Apply the learning algorithm to training set, generating a hypothesis h^*. (3M) ✓ Measure the percentage of example in the test set that is correctly classified by h^*. (3M) ✓ Repeat step 1 to 4 for different size of training sets and different randomly selected training sets of each size. (2M)
	UNIT IV SOFTWARE AGENTS
	Architecture for Intelligent Agents – Agent communication – Negotiation and Bargaining – Argumentation among Agents – Trust and Reputation in Multi-agent systems.
	<p>Define state-space search. BTL1</p> <p>The most straightforward approach is to use state-space search. Because the descriptions of actions in a planning problem specify both preconditions and effects, it is possible to search in either direction: either forward from the initial state or backward from the goal</p>
	<p>What are the types of state-space search? BTL2</p> <p>The types of state-space search are,</p> <ul style="list-style-type: none"> ✓ Forward state space search; <p>Backward state space search.</p>
	<p>What is Partial-Order Planning? (April/May 2015)(Nov/Dec 2013) BTL2</p> <p>A set of actions that make up the steps of the plan. These are taken from the set of actions in the planning problem. The —empty— plan contains just the Start and Finish actions.</p> <p>Start has no preconditions and has as its effect all the literals in the initial state of the planning problem. Finish has no effects and has as its preconditions the goal literals of the planning problem.</p>
	<p>What are the advantages and disadvantages of Partial-Order Planning? BTL4</p> <p>Advantage: Partial-order planning has a clear advantage in being able to decompose problems into sub problems.</p> <p>Disadvantage: Disadvantage is that it does not represent states directly, so it is harder to estimate how far a partial-order plan is from achieving a goal.</p>
	<p>What is a Planning graph? (Nov/Dec 2011) BTL1</p> <p>A Planning graph consists of a sequence of levels that correspond to time steps in the plan where level 0 is the initial state. Each level contains a set of literals and a set of actions.</p>
	<p>Explain Conditional planning? BTL2</p> <p>Conditional planning is also known as contingency planning, conditional planning deals with incomplete information by constructing a conditional plan that accounts for each possible situation or contingency that could arise</p>
	<p>What is action monitoring? BTL2</p> <p>The process of checking the preconditions of each action as it is executed, rather than checking the preconditions of the entire remaining plan. This is called action monitoring.</p>

	<p>Define planning. BTL1</p> <p>Planning can be viewed as a type of problem solving in which the agent uses beliefs about actions and their consequences to search for a solution.</p>
	<p>List the features of an ideal planner? BTL1</p> <p>The features of an ideal planner are,</p> <ul style="list-style-type: none"> ✓ The planner should be able to represent the states, goals and actions; ✓ The planner should be able to add new actions at any time; <p>The planner should be able to use Divide and Conquer method for solving very big problems.</p>
	PART * B
1	<p>Discuss Knowledge Acquisition. (13M) BTL2</p> <p>Answer:Page. 422-Elaine Rich</p> <p>Knowledge acquisition includes the elicitation, collection, analysis, modeling and validation of knowledge.</p> <ul style="list-style-type: none"> ✓ Issues in Knowledge Acquisition (2M) ✓ Techniques for Knowledge Acquisition(2M). ✓ Protocol-generation techniques(2M) <p>Include many types of interviews (unstructured, semi-structured and structured), reporting and observational techniques.</p> <ul style="list-style-type: none"> ✓ Protocol analysis techniques(2M) <p>Used with transcripts of interviews or text-based information to identify basic knowledge objects within a protocol, such as goals,</p> <ul style="list-style-type: none"> ✓ Hierachy-generation techniques(2M) ✓ Matrix-based techniques(1M) ✓ Sorting techniques(1M) ✓ Limited-information and constrained-processing tasks ✓ Diagram-based techniques(1M)
2	<p>i) Evaluate in detail about knowledge acquisition.(13M) BTL5</p> <p>Answer:Page. 422-Elaine Rich</p> <p>Introduction (3M)</p> <ul style="list-style-type: none"> ✓ The knowledge acquisition component allows the expert to enter their knowledge or expertise into the expert system, and to refine it later as and when required. ✓ The knowledge acquisition process is usually comprised of three principal stages: <p>Knowledge Elicitation (3M)</p> <p>Stages of Knowledge Acquisition (4M)</p> <pre> graph LR A[Identify Problem Characteristics] -- Requirements --> B[Find concepts to represent knowledge] B -- Concepts --> C[Design a structure to organize knowledge] C -- Structure --> D[Formulate rules to embody knowledge] D -- Rules --> E[Validate rules that organise knowledge] C -- Reformulations --> B D -- Redesigns --> C E -- Refinements --> D subgraph IDENTIFICATION [IDENTIFICATION] A end subgraph CONCEPTUALISATION [CONCEPTUALISATION] B end subgraph FORMALISATION [FORMALISATION] C end subgraph IMPLEMENTATION [IMPLEMENTATION] D end subgraph TESTING [TESTING] E end </pre> <p>Levels of Knowledge Analysis (3M)</p> <p>Can you apply the facts to describe?</p> <p>i) Decision tree architecture. (7M)</p>

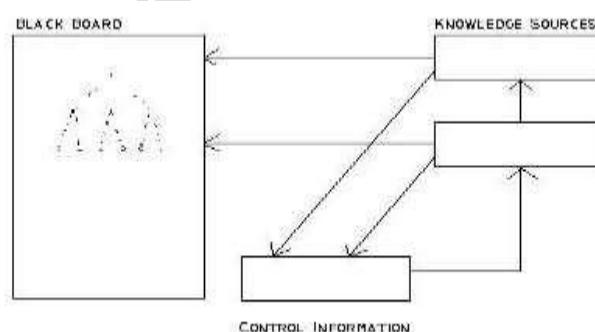
ii) Blackboard system Architecture. (6M) BTL6**Answer:Notes**

Decision tree architecture (4M)

- ✓ Knowledge for ES may be stored in the form of Decision tree.
- ✓ Special tree building editor used.
- ✓ New nodes will be added when additional rules are added.
- ✓ Traversing technique to identify the attributes.
- ✓ Diagram (2M)

**Blackboard system Architecture.**

- ✓ There are 3 components (4M)
- ✓ There are number of *knowledge sources*.
- ✓ A data structure called *Blackboard* which contains current problem state and information needed by the knowledge sources.
- ✓ *Control Information* – Monitors changes in the blackboard.

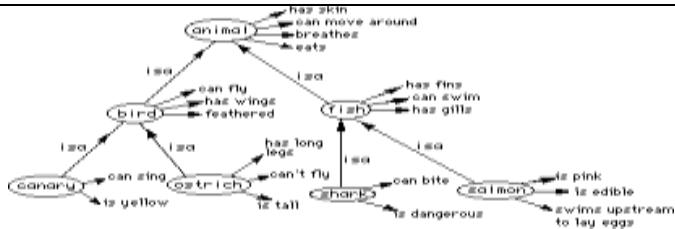
Diagram (2M)**i) Examine about Associative or semantic network architecture. (7M) BTL4****ii) Examine about frame architecture. (6M) BTL4****Answer: Notes**

Associative or semantic network architecture.

Introduction (4M)

- A classic representation technique for propositional information
- Propositions – a form of declarative knowledge, stating facts (true/false)
- Propositions are called –atoms|| – cannot be further subdivided.
- Semantic nets consist of nodes (objects, concepts, situations) and arcs (relationships between them).

Diagram (3M)

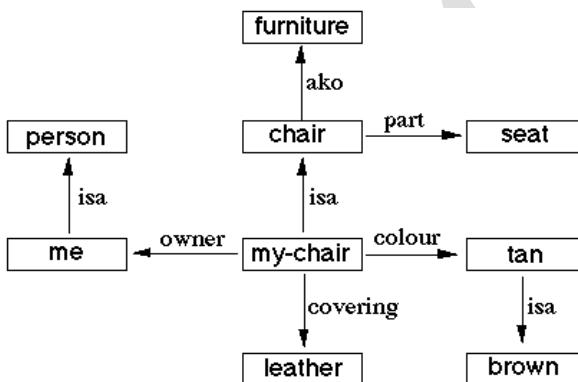


frame architecture

Introduction (4 M)

- One type of schema is a frame (or script – time-ordered sequence of frames).
- Frames are useful for simulating commonsense knowledge.
- Semantic nets provide 2-dimensional knowledge; frames provide 3-dimensional.
- Frames represent related knowledge about narrow subjects having much default knowledge

Diagram (2M)



- (i) How do you examine Performance measure in Knowledge acquisition? (7M) BTL4
(ii) Describe in detail about Characteristic feature of expert system.
(6M) BTL-4

Answer:Page. 422-Elaine Rich

Performance measure in Knowledge acquisition:(7M)

- Expert may not have required knowledge in some areas
- Expert may not be consciously aware of required knowledge needed
- Expert may not be able to communicate the knowledge needed to knowledge engineer Knowledge engineer may not be able to structure knowledge for entry into knowledge base

Characteristic feature of expert system. (6M)

- ✓ simulate human reasoning

- i) perform reasoning over representations

PART * C

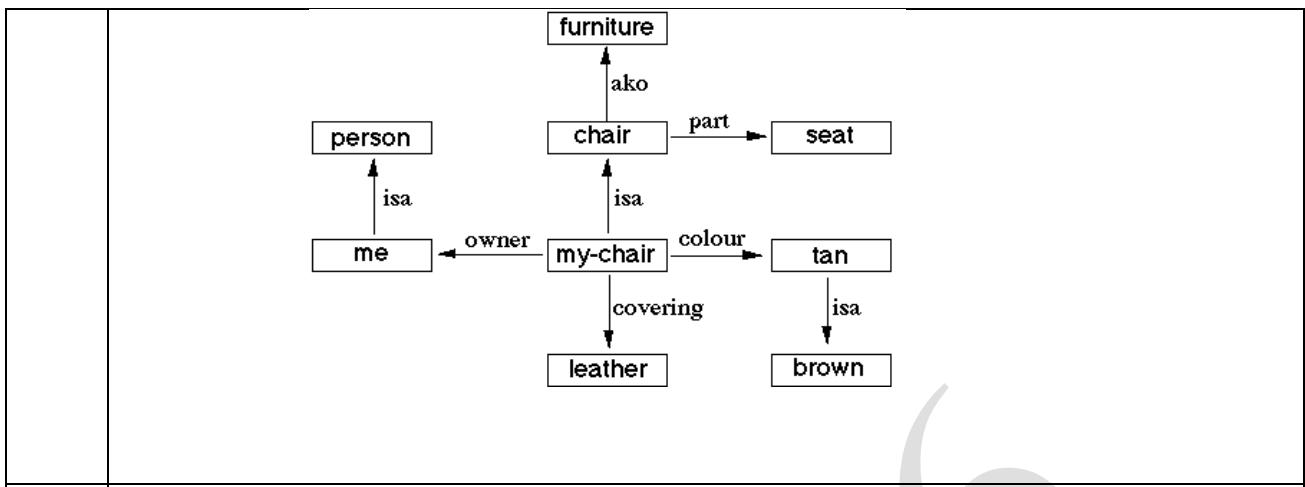
- 1 Explain the need, significance and evolution of XCON expert system. (15M) BTL5
Answer: Notes
XCON

	<ul style="list-style-type: none"> ✓ Expert system to configure VAX -11/780 computers ✓ Developed by collaboration between Carnegie Mellon University and Digital Equipment Corporation ✓ Stages ✓ Identification (1M) ✓ Conceptualization(2M) ✓ Formalization(2M) ✓ Implementation(2M) ✓ Testing and evaluation (2M) ✓ Maintenance(2M) ✓ Domain(1M) ✓ Task(1M) ✓ Input(1M) (iii) Output(1M)
2	<p>Design an expert system for Travel recommendation and discuss its roles.(15M) BTL6</p> <p>Answer: Notes</p> <ul style="list-style-type: none"> ✓ Roles(5M) ✓ Tickets ✓ Source ✓ Destination ✓ Availability checker ✓ updater ✓ Example case study(5M) <p>Explanation (5M)</p>
3	<p>Illustrate in detail about expert system Shells. (13M) BTL5</p> <p>Answer:Page. 424-Elaine Rich</p> <p>Introduction (3M)</p> <ul style="list-style-type: none"> ✓ knowledge base (2M) ✓ the inference mechanism (2M) ✓ working memory(2M) ✓ user interface (2M) ✓ diagram (3M)
4	<p>Explain the architecture of Expert System with neat diagram. (13M) BTL3</p> <p>Answer: Page. 424-Elaine Rich</p> <p>Introduction: diagram (3M)</p> <pre> graph LR Expert((Expert)) <--> KA[Knowledge acquisition system] KA <--> KB[Knowledge base of facts, rules, heuristics] KB <--> IE[Inference engine] IE <--> UI[User interface] User((User)) <--> UI KA <--> KE((Knowledge Engineer)) KB <--> KE IE <--> KE </pre> <p>The diagram illustrates the architecture of an Expert System. It consists of several interconnected components: Knowledge acquisition system, Knowledge base of facts, rules, heuristics, Inference engine, User interface, and a Knowledge Engineer. The Knowledge acquisition system, Knowledge base, and Inference engine are interconnected in a loop. The User interface is connected to the Inference engine. The Knowledge Engineer is connected to all three central components (Knowledge acquisition system, Knowledge base, and Inference engine). Additionally, there are bidirectional connections between the Expert and the Knowledge acquisition system, and between the User and the User interface.</p> <p>Knowledge Base: Knowledge about problem domain in the form of static and dynamic databases. (3M)</p> <p>Inference Engine: It consists of inference mechanism and control strategy. (2M)</p> <p>Knowledge Acquisition: Knowledge acquisition module allows system to acquire knowledge about the problem domain (3M)</p> <p>User Interfaces: Allows user to communicate with system in interactive mode (2M)</p>

	UNIT V APPLICATIONS
	AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing - Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving.
1.	Define an expert system. BTL1 An expert system is a computer program that attempts to mimic human experts by the system's capability to render advice, to teach and execute intelligent tasks.
2.	Define natural language processing. BTL1 Natural language processing is a program that permits (to a certain degree) a human-computer dialogue in a conversational, day-to-day language (a natural language like English, French, or Dutch)
3.	Describe speech recognition and understanding. Why is it useful? BTL1 Speech or voice recognition is a data input method. For example, the computer recognizes and understands one (or a few) word commands. Speech understanding on the other hand is the computer's ability to understand a spoken language. That is, the computer understands the meaning of sentences and paragraphs through syntax and semantics.
4.	Define an intelligent agent. Why is it useful? BTL2 An intelligent agent is a program that runs in the background and learns your patterns, like any other agent working for you. It learns your needs to serve you better. The little paperclip guy that shows up in Microsoft Word is an example of an intelligent agent.
5.	List the major benefits of intelligent computer-aided instruction BTL2 The major benefits are: <ul style="list-style-type: none"> ✓ Individualized, ✓ self-adjusted level of instruction; ✓ immediate feedback; ✓ portability; ✓ consistency <p>Better control of updating and variety of presentations.</p>
6.	Define the ES development environment and contrast it with the consultation environment. BTL1 The development environment includes the activities and support that are necessary to acquire and represent the knowledge as well as to make inferences and provide explanations. The major players in this environment are the knowledge engineer and the domain expert who act as builders. Once the system is completed it is used for consultation by the nonexpert user via the consultation environment.
7.	List and define the major components of an ES. BTL2 The major components are: <ul style="list-style-type: none"> ✓ Knowledge base--the software that represents the knowledge. ✓ Inference engine--the reasoning mechanism. ✓ User interface--the hardware and software that provide the dialogue between people and the computer. ✓ Domain expert--the individual who is considered an expert. ✓ Knowledge engineer--the individual who acquires and represents the knowledge. ✓ Explanation facility--the software that answers questions such as "Why" and —How." ✓ Blackboard--a workplace for storing and working on intermediate information. ✓ Reasoning improvement--a facility (not available commercially) for improving the reasoning capabilities of an ES. <p>User--the non-expert who uses the machine for consultation. Hardware--the hardware that is needed to support the ES.</p>

8.	What is the role of a knowledge engineer? BTL1 Major duties are to acquire and represent the knowledge. Some knowledge engineers do the computer programming as well.
9.	Describe how expert systems perform inference BTL2 The brain of an expert system is the inference engine that provides a methodology for reasoning about information in the knowledge base. Inference can be performed using semantics networks, production rules, and logic statements.
10.	What are the major activities performed in the ES blackboard (workplace)? BTL2 The blackboard records intermediate hypotheses and decisions, devises a plan of how to attack a problem, provides an agenda of actions awaiting execution, and lists the candidate solutions to be examined.
11.	Describe generic categories of ES applications BTL2 Genetic categories of ES applications are: <ul style="list-style-type: none"> ✓ Rule-based ES. Knowledge is represented by a series of rules. ✓ Frame-based systems. Knowledge is represented as a series of frames (an object-oriented approach). ✓ Hybrid systems. Involve several approaches such as fuzzy logic and neural networks. ✓ Model-based systems. Structured around a model that simulates the structure and function of the system under study. ✓ Ready-made systems. Utilize prepackaged software. Real-time systems. Systems designed to produce a just-in-time response.
12.	Describe some of the limitations of ES. BTL1 Knowledge is not always readily available. <ul style="list-style-type: none"> ✓ It can be difficult to extract expertise from humans. There are frequently multiple correct assessments. Time pressures. ✓ Users have cognitive limits. ✓ ES works well only within a narrow domain of knowledge. ✓ Most experts do not have an independent means to validate results. Vocabulary is often limited and difficult to understand. ✓ Help from knowledge engineers is difficult to obtain and costly. Potential for lack of trust on the part of the end-users. Knowledge transfer is subject to biases.
13.	Describe the success factors of ES. BTL1 Success factors are: <ul style="list-style-type: none"> ✓ Level of knowledge must be sufficiently high. ✓ Expertise must be available from at least one expert. ✓ The problem to be solved must be fuzzy. ✓ The problem must be narrow in scope. ✓ The shell must be of high quality and naturally store and manipulate the knowledge. ✓ The user interface must be friendly to novice users. ✓ The problem to be solved must be difficult and important enough to justify the development of a system. ✓ Knowledgeable developers with good people skills are needed. The impact of the ES must be considered. ○ The impact should be favorable. Management support is needed.
16.	What are the benefits of deploying an ES on the Web? BTL1 The expertise is made available to a wider audience, use of a common interface, etc.
17.	How can an ES help a decision maker in Web use? BTL3 An ES can advise a user on how to proceed in doing his/her work (Web searches, database access, etc.).

18.	<p>What is meta knowledge? How meta knowledge is represents in rule-based expert systems? AN (MAY/JUNE2016) BTL2</p> <p>Meta Knowledge: Knowledge about knowledge Meta Knowledge is represented in rule-based expert systems as meta rules.</p>
19.	<p>Write any four earliest expert systems. (MAY/JUNE2016) BTL1</p> <p>Four earliest expert systems: DENDRAL,MYCIN,PROSPECTOR,XCON</p>
	PART * B
1	<p>How did you describe Rule-Based system Architecture? (13M) BTL-3</p> <p>Answer:Notes</p> <p>Introduction : The production rules as knowledge representation mechanism are used in the design of many "Rule-based systems" also called "Production systems"(2M)</p> <p>Types of rules (6M)</p> <ul style="list-style-type: none"> ■ Knowledge Declarative Rules: ■ Inference ProceduralRules ■ Metarules <p>Examples and explanationof production rules : (5M)</p> <ul style="list-style-type: none"> - IF condition THEN action - IF premise THENconclusion <p>IF proposition p1 and proposition p2 are true THEN proposition p3 istrue</p>
2	<p>Examine about Associative or semantic network architecture. (7M) BTL4</p> <p>Examine about frame architecture. (6M) BTL4</p> <p>Answer: Notes</p> <p>Associative or semantic network architecture.</p> <p>Introduction (4M)</p> <ul style="list-style-type: none"> • A classic representation technique for propositional information • Propositions – a form of declarative knowledge, stating facts (true/false) • Propositions are called -atoms – cannot be further subdivided. • Semantic nets consist of nodes (objects, concepts, situations) and arcs (relationships between them). <p>Diagram (3M)</p> <p style="text-align: center;">frame architecture</p> <p>Introduction (4 M)</p> <ul style="list-style-type: none"> • One type of schema is a frame (or script – time-ordered sequence of frames). • Frames are useful for simulating commonsense knowledge. • Semantic nets provide 2-dimensional knowledge; frames provide 3-dimensional. • Frames represent related knowledge about narrow subjects having much default knowledge <p>Diagram (2M)</p>



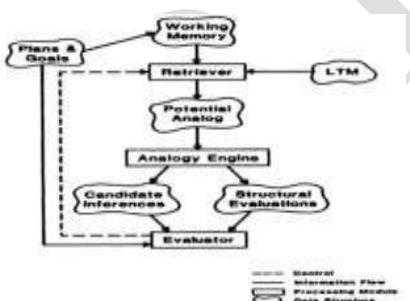
- 3 ii) What conclusion can you infer from Analogical reasoning Architecture? (6M) BTL3
 iii) Explain in detail about Neural Network Architecture? (7M) BTL4

Answer: Page. 379-Elaine Rich

Analogical reasoning Architecture(4M)

Analogy-based reasoning: This term is sometimes used, as a synonym to case-based reasoning, to describe the typical case-based approach

Diagram (2M)



Neural Network Architecture (4M)

Examples one neural neural f production rules :

- IF condition THEN action
- IF premise THEN conclusion
- IF proposition p₁ and proposition p₂ are true THEN proposition p₃ is true

Explanation:

- 4 i) Evaluate in detail about knowledge acquisition.(13M) BTL5

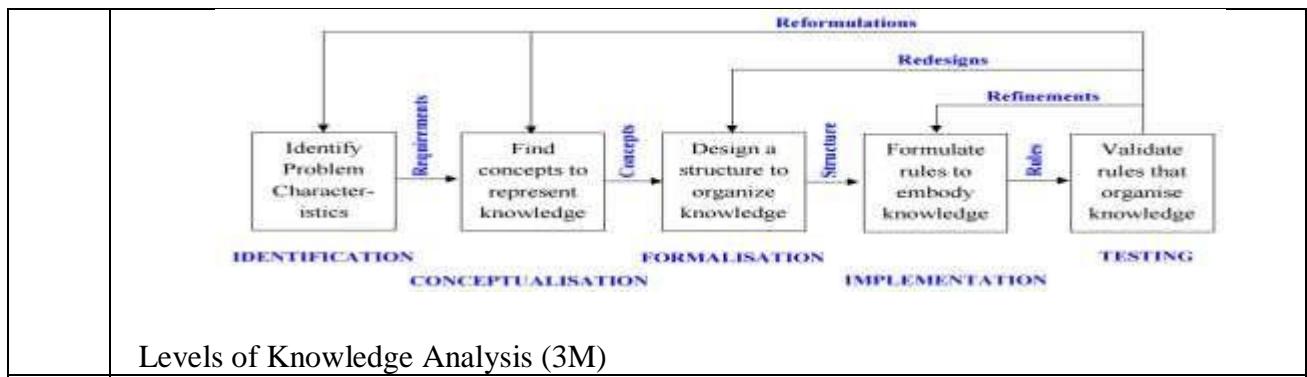
Answer: Page. 422-Elaine Rich

Introduction (3M)

- ✓ The knowledge acquisition component allows the expert to enter their knowledge or expertise into the expert system, and to refine it later as and when required.
- ✓ The knowledge acquisition process is usually comprised of three principal stages:

Knowledge Elicitation (3M)

Stages of Knowledge Acquisition (4M)

**Discuss Knowledge Acquisition. (13M) BTL2****Answer:Page. 422-Elaine Rich**

Knowledge acquisition includes the elicitation, collection, analysis, modeling and validation of knowledge.

- ✓ Issues in Knowledge Acquisition (2M)
- ✓ Techniques for Knowledge Acquisition(2M)
- ✓ Protocol-generation techniques(2M)

Include many types of interviews (unstructured, semi-structured and structured), reporting and observational techniques.

- ✓ Protocol analysis techniques(2M)

Used with transcripts of interviews or text-based information to identify basic knowledge objects within a protocol, such as goals,

- ✓ Hierarchy-generation techniques(2M)
- ✓ Matrix-based techniques(1M)
- ✓ Sorting techniques(1M)
- ✓ Limited-information and constrained-processing tasks

Diagram-based techniques(1M)

PART * C**1 Explain the need, significance and evolution of ICON expert system.(15M) BTL5****Answer: Notes****XCON**

- ✓ Formalization(2M)
- ✓ Implementation(2M)
- ✓ Testing and evaluation (2M)
- ✓ Maintenance(2M)
- ✓ Domain(1M)
- ✓ Task(2M)
- ✓ Input(2M)
- ✓ Output(2M)

2 Design an expert system for Travel recommendation and discuss its roles.(15M) BTL6**Answer: Notes**

- ✓ Roles(5M)
- ✓ Example case study(5M)
- ✓ Explanation (5M)

CS8601

MOBILE COMPUTING**L T P C
3 0 0 3****OBJECTIVES:****The student should be made to:**

- To understand the basic concepts of mobile computing.
- To learn the basics of mobile telecommunication system .
- To be familiar with the network layer protocols and Ad-Hoc networks.
- To know the basis of transport and application layer protocols.
- To gain knowledge about different mobile platforms and application development.

UNIT I - INTRODUCTION**9**

Introduction to Mobile Computing – Applications of Mobile Computing- Generations of Mobile Communication Technologies- Multiplexing – Spread spectrum -MAC Protocols – SDMA- TDMA- FDMA- CDMA

UNIT II - MOBILE INTERNET PROTOCOL AND TRANSPORT LAYER**9**

Introduction to Cellular Systems – GSM – Services & Architecture – Protocols – Connection Establishment – Frequency Allocation – Routing – Mobility Management – Security – GPRS- UMTS – Architecture – Handover – Security

UNIT III - MOBILE TELECOMMUNICATION SYSTEM**9**

Mobile IP – DHCP – AdHoc– Proactive protocol-DSDV, Reactive Routing Protocols – DSR, AODV , Hybrid routing –ZRP, Multicast Routing- ODMRP, Vehicular Ad Hoc networks (VANET) –MANET Vs VANET – Security.

UNIT IV - MOBILE AD-HOC NETWORKS**9**

Mobile TCP– WAP – Architecture – WDP – WTLS – WTP –WSP – WAE – WTA Architecture – WML

UNIT V - MOBILE PLATFORMS AND APPLICATIONS**9**

Mobile Device Operating Systems – Special Constraints & Requirements – Commercial Mobile Operating Systems – Software Development Kit: iOS, Android, BlackBerry, Windows Phone – MCommerce – Structure – Pros & Cons – Mobile Payment System – Security Issues

TOTAL: 45 PERIODS**OUTCOMES:****At the end of the course, the student should be able to:**

- Explain the basics of mobile telecommunication systems
- Illustrate the generations of telecommunication systems in wireless networks
- Determine the functionality of MAC, network layer and Identify a routing protocol for a given Ad hoc network
- Explain the functionality of Transport and Application layers
- Develop a mobile application using android/blackberry/ios/Windows SDK

TEXT BOOK:

- 1.Jochen Schiller, —Mobile Communications‖, PHI, Second Edition, 2003.
2. Prasant Kumar Pattnaik, Rajib Mall, “Fundamentals of Mobile Computing”, PHI Learning Pvt. Ltd, New Delhi – 2012

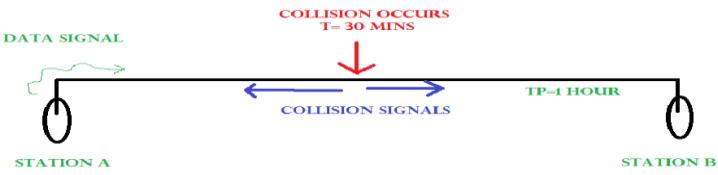
REFERENCES:

1. Dharma Prakash Agarval, Qing and An Zeng, "Introduction to Wireless and Mobile systems", Thomson Asia Pvt Ltd, 2005.
2. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober, “Principles of Mobile Computing”, Springer, 2003.
3. William.C.Y.Lee,“Mobile Cellular Telecommunications-Analog and Digital Systems”, Second Edition,Tata Mc Graw Hill Edition ,2006.
4. C.K.Toh, “AdHoc Mobile Wireless Networks”, First Edition, Pearson Education, 2002.
5. Android Developers : <http://developer.android.com/index.html>
6. Apple Developer : <https://developer.apple.com/>
7. Windows Phone Dev Center : <http://developer.windowsphone.com>
8. BlackBerry Developer : <http://developer.blackberry.com/>

UNIT I INTRODUCTION	
Introduction to Mobile Computing – Applications of Mobile Computing- Generations of Mobile Communication Technologies- Multiplexing – Spread spectrum -MAC Protocols – SDMA- TDMA- FDMA- CDMA	
PART * A	
Q.N o	Questions
1	<p>Define mobile computing. BTL 1</p> <p>It is defined as the capability to change location while communicating to invoke computing services at some remote computers. It is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link.</p>
2	<p>List the advantage of mobile computing. BTL 1 (May/June 2016)</p> <ul style="list-style-type: none"> • Increase in Productivity • Entertainment • Portability • Cloud Computing
3	<p>Give the properties of MAC protocols. BTL 2</p> <ul style="list-style-type: none"> • It should help maximize the utilization of channels • Channel allocation needs to be fair. No node should be discriminated against at any time and made to wait for an unduly long time for transmission.
4	<p>Give some mobile computing applications. BTL 2</p> <ul style="list-style-type: none"> • Emergency services • Vehicles. • CDPD – Cellular Digital Packet Data
5	<p>What is Mobility? BTL 1</p> <p>A person who moves</p> <ul style="list-style-type: none"> • Between different geographical locations • Between different networks • Between different communication devices • Between different applications <p>A device that moves</p> <ul style="list-style-type: none"> • Between different geographical locations • Between different networks
6	<p>What is wireless communication? BTL 1</p> <p>Wireless communication is the transfer of information over a distance without the use of electrical conductors or wires. The distance involved may be short or long lines.</p>

7	<p>List the characteristics of mobile computing. BTL 1</p> <ul style="list-style-type: none"> • Ubiquity • Location awareness • Adaptation • Broadcast • Personalization 										
8	<p>How MAC protocols are classified? BTL 3</p> <p>They are classified into</p> <ul style="list-style-type: none"> • Fixed assignment schemes • Random assignments schemes • Demand – based schemes 										
9	<p>Differentiate between wired network and mobile. BTL 2 (APR/MAY 2017)</p> <table border="1" data-bbox="372 646 1041 952"> <thead> <tr> <th data-bbox="372 646 714 688">Wired network</th><th data-bbox="714 646 1041 688">Mobile network</th></tr> </thead> <tbody> <tr> <td data-bbox="372 688 714 730">1. High bandwidth</td><td data-bbox="714 688 1041 730">1. Low Bandwidth</td></tr> <tr> <td data-bbox="372 730 714 804">2. High power machines</td><td data-bbox="714 730 1041 804">2. Low power machines</td></tr> <tr> <td data-bbox="372 804 714 878">3. Can listen on wire</td><td data-bbox="714 804 1041 878">3. Hidden terminal problem</td></tr> <tr> <td data-bbox="372 878 714 952">4. Connected operation</td><td data-bbox="714 878 1041 952">5. Disconnected operation</td></tr> </tbody> </table>	Wired network	Mobile network	1. High bandwidth	1. Low Bandwidth	2. High power machines	2. Low power machines	3. Can listen on wire	3. Hidden terminal problem	4. Connected operation	5. Disconnected operation
Wired network	Mobile network										
1. High bandwidth	1. Low Bandwidth										
2. High power machines	2. Low power machines										
3. Can listen on wire	3. Hidden terminal problem										
4. Connected operation	5. Disconnected operation										
10	<p>What are the functions of mobile computing? BTL 1</p> <ul style="list-style-type: none"> • Session mobility • Device mobility • Service mobility • Host mobility 										
11	<p>State the issues of wireless MAC protocols. BTL 1</p> <ul style="list-style-type: none"> • Hidden terminal problem • Exposed • Near & Far 										
12	<p>Give some examples for fixed assignment and random Assignment schemes. BTL 1</p> <p>FAS- FDMA, TDMA, CDMA RAS – Aloha and CSMA.</p>										
13	<p>What is the advantage of TDMA? BTL 1</p> <ul style="list-style-type: none"> • Flexible bit rate • No frequency guard band required • Extended battery life • Easy for mobile or base stations to initiate and execute hand off 										
14	<p>What is the disadvantage of using FDMA? BTL 1</p> <ul style="list-style-type: none"> • The presence of guard signals. • Maximum bit rate per channel is fixed. • Requires right RF filtering to minimize adjacent channel interference. 										

15	<p>List various Random Assignment schemes in MAC. BTL 1 (Nov/dec2016)</p> <ul style="list-style-type: none"> • ALOHA • Slotted ALOHA • CSMA • CSMA/CD • CSMA/CA
16	<p>What are the limitations of Mobile Computing? BTL 1 (Nov/dec2016)</p> <p>Wireless Medium</p> <ul style="list-style-type: none"> • Cost of Networks • Quantity and reliability of bandwidth • Environment obstacles <p>Portability -Mobile Restrictions</p> <ul style="list-style-type: none"> • Low Resources • Battery Constraint
17	<p>Why do Hidden and Exposed terminal problem arise. BTL 1 (May/June 2016)</p> <p>Hidden terminal problem is due to the fact that a node (say A) transmitting to another node (say B) cannot hear transmissions from another node C, which might also be transmitting to B, and might interfere with the A-to-B transmissions.</p> <p>Exposed node problem occurs when a node is prevented from sending packets to other nodes because of a neighboring transmitter.</p>
18	<p>Show that Barker code has good auto correlation. BTL 3</p> <p>When the receiver attempts to correlate the received coded symbols with respect to any of the codes which it internally generates, it is not able to correlate even when it uses exactly the same code as the one used for transmission.</p> <ul style="list-style-type: none"> • Reasons for no correlation • Propagation delay • Inappropriate code
19	<p>Give the difference between 1G, 2G, 2.5G, 3G mobile network communications. BTL 1</p> <p>1G –Voice -only communication.</p> <p>2G –Communicate voice as well as data signals.</p> <p>2.5G–Enhancements of the second generation and sport data rates up to 100 kpbs.</p> <p>3G –Mobile devices communicate at even higher data rates and support voice, data, and multimedia streams. High data rates in 3G devices enable transfer of video clips and faster multimedia communication.</p>
20	<p>What are the basic services provided by the MAC layer? BTL 1</p> <ul style="list-style-type: none"> • Asynchronous data service (mandatory)

	<ul style="list-style-type: none"> • Time-bounded service (optional)
21	<p>Define Mobile Binding. BTL 1 A binding created for providing mobility to a mobile node after registration at a foreign network.</p>
22	<p>What is MAC? BTL 1 Message authentication codes (MAC) are used to authenticate messages during transmission. MAC of a message is created using a cryptographic MAC function which is similar to the hash function but has different security requirements.</p>
23	<p>Define MACA Protocol. BTL 1 Multiple Access with Collision Avoidance (MACA) is a slotted media access control protocol used in wireless LAN data transmission to avoid collisions caused by the hidden station problem and to simplify exposed station problem.</p>
24	<p>Define Collision Detection based protocol for wireless networks. BTL 1 CSMA/CD (Carrier Sense Multiple Access/ Collision Detection) is a media-access control method widely used in Ethernet technology/LANs.</p> 
25	<p>Compose a role which is played by Radio/Infrared signals play in Mobile Computing. BTL 6</p> <ul style="list-style-type: none"> • Radio transmission uses radio-wave frequencies to send data directly between transmitters and receivers. • Infrared light -red light that is not commonly visible to human eyes. Red lights are used in remote controls.
PART * B	
Q.N 0	Questions
1	<p>Describe in detail about characteristics of mobile computing. (13M) (Nov/Dec2016)</p> <p align="right">BTL 4</p> <p>Answer: Page:28-31 - Prasant Kumar Pattnaik</p> <p>Definition: (2M)</p> <p>A computing environment is said to be mobile, when either the sender or the receiver of information can be on the move while transmitting or receiving information.</p> <p>Explanation (10M)</p> <p>Ubiquity: (2M)</p> <p>Ability of a user - perform computations from anywhere - at any time.</p>

	<p>Location awareness: (2M)</p> <p>Hand held device equipped with global positioning system (GPS) - transparently provide information - current location of a user - tracking system.</p> <p>Adaptation: (2M)</p> <p>Ability of a system - adjust to bandwidth fluctuation - without inconveniencing user.</p> <p>Broadcast: (2M)</p> <p>Broadcast nature - underlying communication network - mobile computing environment, efficient delivery of data - made simultaneously to hundreds of mobile users.</p> <p>Personalization: (2M)</p> <p>Services in mobile environment - easily personalized - user's profile</p>
2	<p>Explain different layers of structure of mobile computing Application. (13M) (May/June 2016) BTL 1</p> <p>Answer: Page: 88- 110 - Prasant Kumar Pattnaik</p> <p>Definition: (2M)</p> <p>A mobile computing application is usually structured in terms of the functionalities implemented.</p> <p>Explanation: (6M)</p> <p>Presentationtier, Application tier Data tier.</p> <pre> graph TD A[Find the sales total] --> B[Find the list of sales made in last year] B --> C[Query] B --> D["Sales1 Sales2 Sales3"] C --> E[Database] E --> F[Storage] </pre> <p>Diagram(5M)</p>
3	<p>Write a short on the concepts of</p> <p>FDMA (3M)</p> <p>TDMA (2M)</p> <p>CDMA. (2M)</p> <p>SDMA. (2M) BTB 2 (APR/MAY 2017)</p> <p>Answer: Page: 56 -59 - Prasant Kumar Pattnaik</p>

Explanation (9M)**Fixed Assignment Schemes:**

Important categories of fixed assignment MAC protocols:

- Frequency Division Multiple Access(FDMA)
- Time Division Multiple Access(TDMA)
- Code Division Multiple Access(CDMA)
- Spatial division multiple access (SDMA)

FDMA:

In FDMA - available bandwidth (frequency range) - divided into many narrower frequency bands called channels.

TIME DIVISION MULTIPLE ACCESS (TDMA):

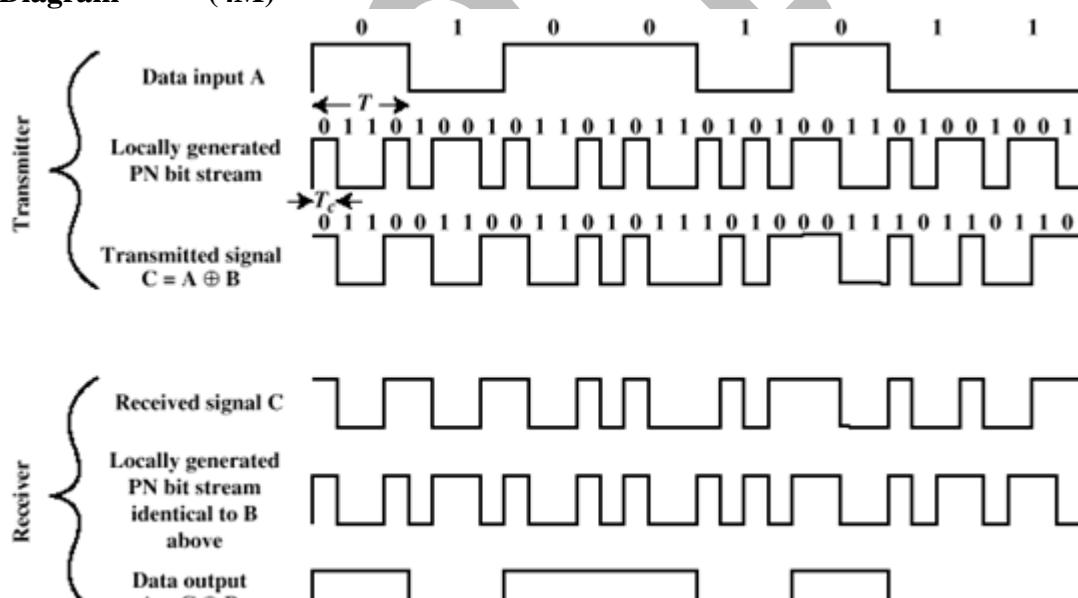
TDMA - access method - multiple nodes - allotted different time slots - access the same physical channel - timeline divided into fixed time slots - divided among multiple nodes that can transmit.

CDMA:

In CDMA - multiple users are allotted different codes - consist of sequences of 0 and 1 to access the same channel.

SDMA:

Spatial division multiple access (SDMA) - channel access method - mobile communication systems - reuses the same set of cell phone frequencies - given service area

Diagram (4M)

4

Discuss in detail about Random assignment scheme and Reservation based schemes. (13M) BTL 2

Answer: Page: 59-61 - Prasant Kumar Pattnaik

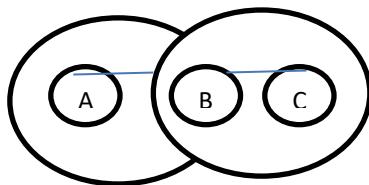
Random Assignment Schemes: (7M)

ALOHA

Slotted ALOHA

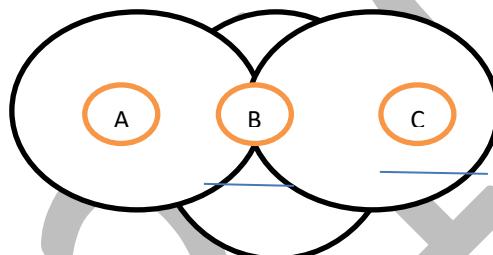
	<p>CSMA CSMA/CD CSMA/CA</p> <p>ALOHA Scheme: Simple communication scheme - developed at the University of Hawai - (also called pure) ALOHA scheme, is a simple protocol.</p> <p>Slotted ALOHA: slotted ALOHA scheme - chances of collisions are attempted - reduced by enforcing restrictions</p> <p>CSMA a node senses - medium before starting to transmit - senses that some transmission -already underway - it defers its transmission.</p> <p>Reservation- based schemes: (6M) A basic form of the reservation scheme is the RTS/CTS scheme. In an RTS/CTS scheme, a sender transmits an RTS (Ready to Send) packet to the receiver before the actual data transmission.</p> <p>MACA MACA - Multiple Access Collision Avoidance. MACA solves - hidden/exposed terminal regulating - transmitter power.</p>
5	<p>Describe in detail about MAC protocols issues. (13M) BTL 4 (APR/MAY 2017)</p> <p>Answer: Page:207 -209 - Prasant Kumar Pattnaik</p> <p>Explanation: (8M)</p> <p>Hidden and exposed terminal problems in infrastructure less Network: (4M)</p> <ul style="list-style-type: none"> Consider three mobile phones A,B, C. The transmission range of A reaches B, but not C (the detection range does not reach C either). The transmission range of C reaches B, but not A. Finally, the transmission range of B reaches A and C, i.e., A cannot detect C and vice versa. A starts sending to B, C does not receive this transmission. C also wants to send something to B and senses the medium. The medium appears to be free, the carrier sense fails. C also starts sending causing a collision at B. But A cannot detect this collision at B and continues with its transmission. A is hidden for C and vice versa. While hidden terminals may cause collisions, the next effect only causes unnecessary delay. Now consider the situation that B sends something to A and wants to transmit data to some other mobile phone outside the interference ranges of A and B. C senses the carrier and detects that the carrier is busy (B's signal). C postpones its transmission until it detects the medium as being idle again. But as A is outside the interference range of C, waiting is not necessary. Causing a 'collision' at B

- does not matter because the collision is too weak to propagate to A.
- In this situation, C is **exposed** to B.



Near and Far terminals: (4m)

- A and B are both sending with the same transmission power. As the signal strength decreases proportionally to the square of the distance, B's signal drowns out A's signal. As a result, C cannot receive A's transmission.
- C as being an arbiter for sending rights (e.g., C acts as a base station coordinating media access). In this case, terminal B would already drown out terminal A on the physical layer. C in return would have no chance of applying a fair scheme as it would only hear B.



- The **near/far effect** is a severe problem of wireless networks using CDM. All signals should arrive at the receiver with more or less the same strength.
- Even if the senders were separated by code, the closest one would simply drown out the others.
- Precise power control is needed to receive all senders with the same strength at a receiver. For example, the UMTS system adapts power 1,500 times per second.

Diagram (5M)

6 Differentiate between FDMA, TDMA, and CDMA. (13M) BTL 2

Answer: Page: 56-59 - Prasant Kumar Pattnaik

Explanation(13M)

FDMA	TDMA	CDMA
Frequency Division Multiple Access or FDMA is an access technology that is used by radio "multiple access" implies the sharing of the resource amongst users, and the "frequency division" describes how the	Time division multiple access (TDMA) is a channel access method shared systems to share the radio spectrum. The terminology	CDMA is a form of "spread-spectrum" signaling, since the shared medium (usually radio) networks.

	sharing is done: by allocating users with different carrier frequencies of the radio spectrum.																																
	Frequency limited	Is Bandlimited system	Power limited system																														
	Single frequency is used for single call	Multiple frequencies are used for multiple calls	Single frequency is used for multiple calls																														
	Filtering in the frequency domain.	Synchronization in time domain	Code plus special receivers.																														
	Cell Capacity is limited.	Cell Capacity is limited.	No absolute limit on channel capacity but it is an interference limited system																														
	Simple, established, robust	Established fully digital, flexible	Flexible, less frequency planning, soft handover																														
	Inflexible, frequencies are scarce resources.	a Guard space needed (multipath propagation) Synchronization needed.	Complex receivers need more powerful control for senders.																														
	Transmission scheme is Continuous	Transmission scheme is Discontinuous	Transmission scheme is Discontinuous																														
	PART * C																																
1	Explain the distinguishing features of various wireless network generations. (15M) (NOV/DEC 2016) Answer: Page: 17-19- Prasant Kumar Pattnaik Explanation (15M)																																
	<table border="1"> <thead> <tr> <th>Generation</th><th>Period of commercial use</th><th>Features</th><th>Standards</th><th>Data speed</th></tr> </thead> <tbody> <tr> <td>1G</td><td>70s to 90s</td><td>Analog transmissions, primarily usage restricted to voice communication</td><td>NMT, AMPS, TACS</td><td>No direct Support</td></tr> <tr> <td>2g</td><td>90s to 2000</td><td>Digital transmissions, improved performance by letting multiple users share a single channel</td><td>GSM</td><td>9.6Kps</td></tr> <tr> <td>2.5G</td><td>2001-2005</td><td>Enhanced multimedia and streaming video, web browsing</td><td>GPRS</td><td>28kps or higher</td></tr> <tr> <td>3G</td><td>2005-2015</td><td>Enhanced multimedia and streaming video capabilities</td><td>UMTS, HSPA, EDGE, W-CDMA</td><td>384 kps or higher</td></tr> <tr> <td>4G</td><td>2010- present</td><td>Support interactive multimedia, voice, video, wireless internet and other broadband services</td><td>LTE, WIMAX</td><td>100 mbps or higher</td></tr> </tbody> </table>			Generation	Period of commercial use	Features	Standards	Data speed	1G	70s to 90s	Analog transmissions, primarily usage restricted to voice communication	NMT, AMPS, TACS	No direct Support	2g	90s to 2000	Digital transmissions, improved performance by letting multiple users share a single channel	GSM	9.6Kps	2.5G	2001-2005	Enhanced multimedia and streaming video, web browsing	GPRS	28kps or higher	3G	2005-2015	Enhanced multimedia and streaming video capabilities	UMTS, HSPA, EDGE, W-CDMA	384 kps or higher	4G	2010- present	Support interactive multimedia, voice, video, wireless internet and other broadband services	LTE, WIMAX	100 mbps or higher
Generation	Period of commercial use	Features	Standards	Data speed																													
1G	70s to 90s	Analog transmissions, primarily usage restricted to voice communication	NMT, AMPS, TACS	No direct Support																													
2g	90s to 2000	Digital transmissions, improved performance by letting multiple users share a single channel	GSM	9.6Kps																													
2.5G	2001-2005	Enhanced multimedia and streaming video, web browsing	GPRS	28kps or higher																													
3G	2005-2015	Enhanced multimedia and streaming video capabilities	UMTS, HSPA, EDGE, W-CDMA	384 kps or higher																													
4G	2010- present	Support interactive multimedia, voice, video, wireless internet and other broadband services	LTE, WIMAX	100 mbps or higher																													
2	Classify the different categories of MAC protocols. Identify the situations under which protocols from one category would be preferable over the other categories. Explain the																																

	<p>working of a reservation-based MAC protocols. (15M)</p> <p>Answer: Page: 61-63- Prasant Kumar Pattnaik</p> <p>Explanation (10M)</p> <p>They are classified into</p> <ul style="list-style-type: none"> • Fixed assignment schemes • Random assignments schemes • Demand – based schemes <p>Random Assignment Schemes: (5M)</p> <p>ALOHA Slotted ALOHA CSMA CSMA/CD CSMA/CA</p> <p>Reservation- based schemes: (5M)</p> <p>A basic form of the reservation scheme is the RTS/CTS scheme. In an RTS/CTS scheme, a sender transmits an RTS (Ready to Send) packet to the receiver before the actual data transmission.</p> <p>MACA MACA - Multiple Access Collision Avoidance. MACA solves - hidden/exposed terminal regulating - transmitter power.</p> <p>Diagram (5M)</p>	<p>BTL 4 (MAY/JUNE</p> <pre> graph LR A((A)) ---[20] B((B)) A ---[42] C((C)) A ---[30] D((D)) B ---[35] D C ---[12] D </pre>
3	<p>Differentiate infrastructure-based networks and infrastructure-less networks with the help of suitable schematic diagrams. (15M)</p> <p>Answer: Page: 63-65 - Prasant Kumar Pattnaik</p> <p>Explanation : (10M)</p> <p>Infrastructure less wireless network - network of mobile nodes without having any central controller.</p> <p>Compared to ad-hoc wireless networks - infrastructure offers advantage of scale, centralized security management, and improved reach.</p> <p>Wireless devices can connect to resources on a wired LAN - which is common business settings - more access points can be added - improve congestion and broaden the reach of the network.</p> <p>Diagram (5M)</p>	<p>BTL 4</p>

Subject Code: CS 8601
Subject Name: Mobile Computing

Year / Sem : III / 6
Subject Handler: Ms. Suganya M

UNIT II MOBILE INTERNET PROTOCOL AND TRANSPORT LAYER

Introduction to Cellular Systems – GSM – Services & Architecture – Protocols – Connection Establishment – Frequency Allocation – Routing – Mobility Management – Security – GPRS- UMTS – Architecture – Handover – Security

PART * A

<p>1</p> <p>List the features of Mobile IP.</p> <ul style="list-style-type: none"> • Transparency • Compatibility • Security • Efficiency and scalability 	108 BTL 1
<p>2</p> <p>What are the four types of handover available in GSM? BTL 1</p> <ul style="list-style-type: none"> • Intra cell Handover • Inter cell Intra BSC Handover • Inter BSC Intra MSC handover • Inter MSC Handover 	
<p>3</p> <p>How do I - TCP isolate problems on the wireless link? BTL 3</p> <p>I -TCP isolate problems on the wireless link:</p> <ul style="list-style-type: none"> • I. TCP splits the connection into two parts .a wired / fixed and a wireless / mobile part. • I - TCP isolate problems on the wireless link from the fixed networks. 	
<p>4</p> <p>List advantages of mobile TCP. BTL 1</p> <ul style="list-style-type: none"> • It maintains the TCP end .to .end semantics. The SH does not sent any ACK itself but forwards the ACKs from the MH. • If the MH is disconnected, it avoids useless retransmissions, slow starts or breaking connections by simply shrinking the sender's window to 0. • Since it does not buffer data in the SH as I-TCP does, it is not necessary to forward buffers to a new SH. Lost packets will be automatically retransmitted to the new SH. 	
<p>5</p> <p>List disadvantages of mobile TCP. BTL 1</p> <p>As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption. A modified TCP on the wireless link not only requires modification to the MH protocol software but also now new network elements like the bandwidth manager.</p>	

6	Define fast retransmit. BTL 1 The gap in the packet stream is not due to severe congestion, but a simple packet loss due to a transmission error. The sender can now retransmit the missing packet before the timer expires. This behaviour is called fast retransmit.
7	Define COA. BTL 1 (NOV/DEC 2016) Care –of address is the address of the current tunnel end point for the Mobile node. It gives us the actual location of the MN from an IP point of view. Can be chosen e.g via DHCP.
8	What are the types of COA? BTL 1 <ul style="list-style-type: none"> • Foreign agent COA • Collacated COA
9	What are the four messages transmitted in optimized mobile IP? BTL 1 <ul style="list-style-type: none"> • Binding request • Binding acknowledgement • Binding update • Binding warning
10	What are the features of mobile IP? BTL 1 <ul style="list-style-type: none"> • Transparency • Compatibility • Security • Efficiency and scalability
11	What are the key mechanisms used in Mobile IP? BTL 1 <ul style="list-style-type: none"> • Discovering the care-of- address • Registering the care-of- address • Tunneling the care-of- address
12	List the use route optimization. BTL 1 (APR/MAY 2017) <ul style="list-style-type: none"> • Enable direct notification of the corresponding host • Direct tunneling from the corresponding host to the mobile host • Binding cache maintained at the corresponding host

13	<p>Illustrate the mechanisms used by DHCP for IP address allocation. BTL 3</p> <ul style="list-style-type: none"> • Automatic allocation • Dynamic allocation • Manual allocation
14	<p>Define GPRS. BTL 1 The General Packet Radio Service provides packet mode transfer for applications that exhibit traffic patterns such as frequent transmission of small volumes.</p>
15	<p>List out the service of GSM. BTL 1 (NOV/DEC 2016) The key advantages of GSM systems to consumers have been higher voice quality and low cost alternatives to making calls, such as the Short Message Service. The advantages for network operator have been the ease of deploying equipments from any vendors that implement the standard. Like other cellular standards, GSM allows network operators to offer roaming services so that subscribers can use their phones on GSM networks all over the world</p>
16	<p>Why routing in multi hop adhoc networks are complicated? BTL 1 (APR/MAY 2017) Routing is complicated because of frequent topology changes, different capabilities of the nodes, varying propagation characteristics. Further, no control instance can support routing.</p>
17	<p>What is the basic purpose of DHCP? BTL 1 (MAY/JUNE 2016)</p> <ul style="list-style-type: none"> • DHCP is mainly used to simplify the installation and maintenance of networked computer • DHCP is a mechanism for configuring nodes, parameters acquired via DHCP are eg., IP address, default gateway, DNS server, subnet mask, etc.
18	<p>Define Tunneling and Encapsulation. BTL 1 (MAY/JUNE 2016)</p> <ul style="list-style-type: none"> • Encapsulation: is the mechanism of taking a packet consisting of packet header and data and putting it into the data part of a new packet. • Tunnel: establishes a virtual pipe for data packet between a tunnel entry and a tunnel endpoint.
19	<p>What are the three types of encapsulation? BTL 1</p> <ul style="list-style-type: none"> • IP - in - IP Encapsulation • Minimal Encapsulation • Generic Routing Encapsulation
20	<p>State the use of BOOTP Protocol. BTL 3 (NOV/DEC 2016) The Bootstrap Protocol(BOOTP) is a computer networking protocol used in Internet Protocol networks to automatically assign an IP address to network devices from a configuration server. The BOOTP was originally defined in RFC 951.</p>
21	<p>What is the need for encapsulation? BTL 1</p> <ul style="list-style-type: none"> • To hide the original header information

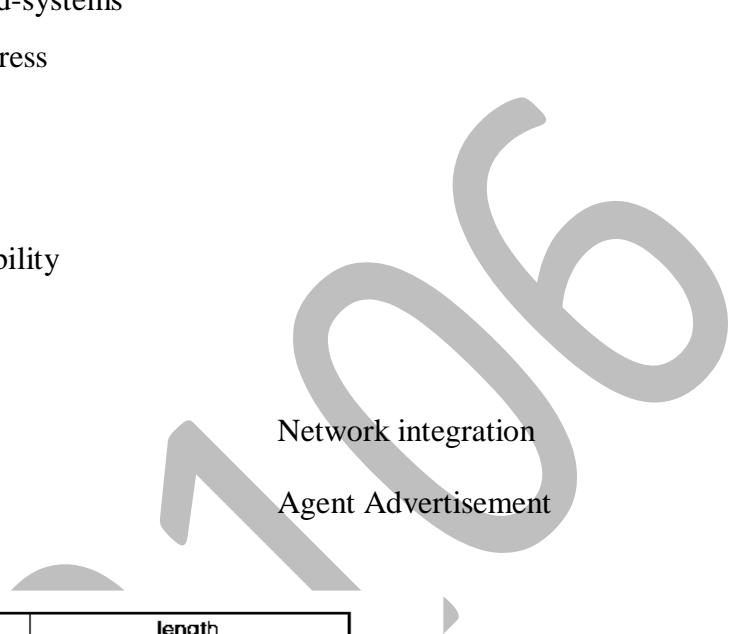
	<ul style="list-style-type: none"> • To provide data independence
22	<p>State the IP datagram structure. BTL 2</p> <p>The IP datagram structure. An IP datagram consists of a header part and text part. The header has a 20 bytes fixed part and a variable length optional part. It is transmitted in big endian order.</p>
23	<p>Define congestion avoidance. BTL 1</p> <p>Transmission Control Protocol (TCP) uses a network congestion-avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease (AIMD) scheme, with other schemes such as slow start and congestion window to achieve congestion avoidance.</p>
24	<p>What is Care of address? BTL 1</p> <p>Used in Internet routing, a care-of address (usually referred to as CoA) is a temporary IP address for a mobile device. This allows a home agent to forward messages to the mobile device.</p>
25	<p>What are the ways to reduce the congestion in a mobile network? BTL 1</p> <ul style="list-style-type: none"> • Network monitoring • Network Segmentation • <i>Use a Content Delivery Network</i> • <i>Reconfigure TCP/IP Setting.</i>

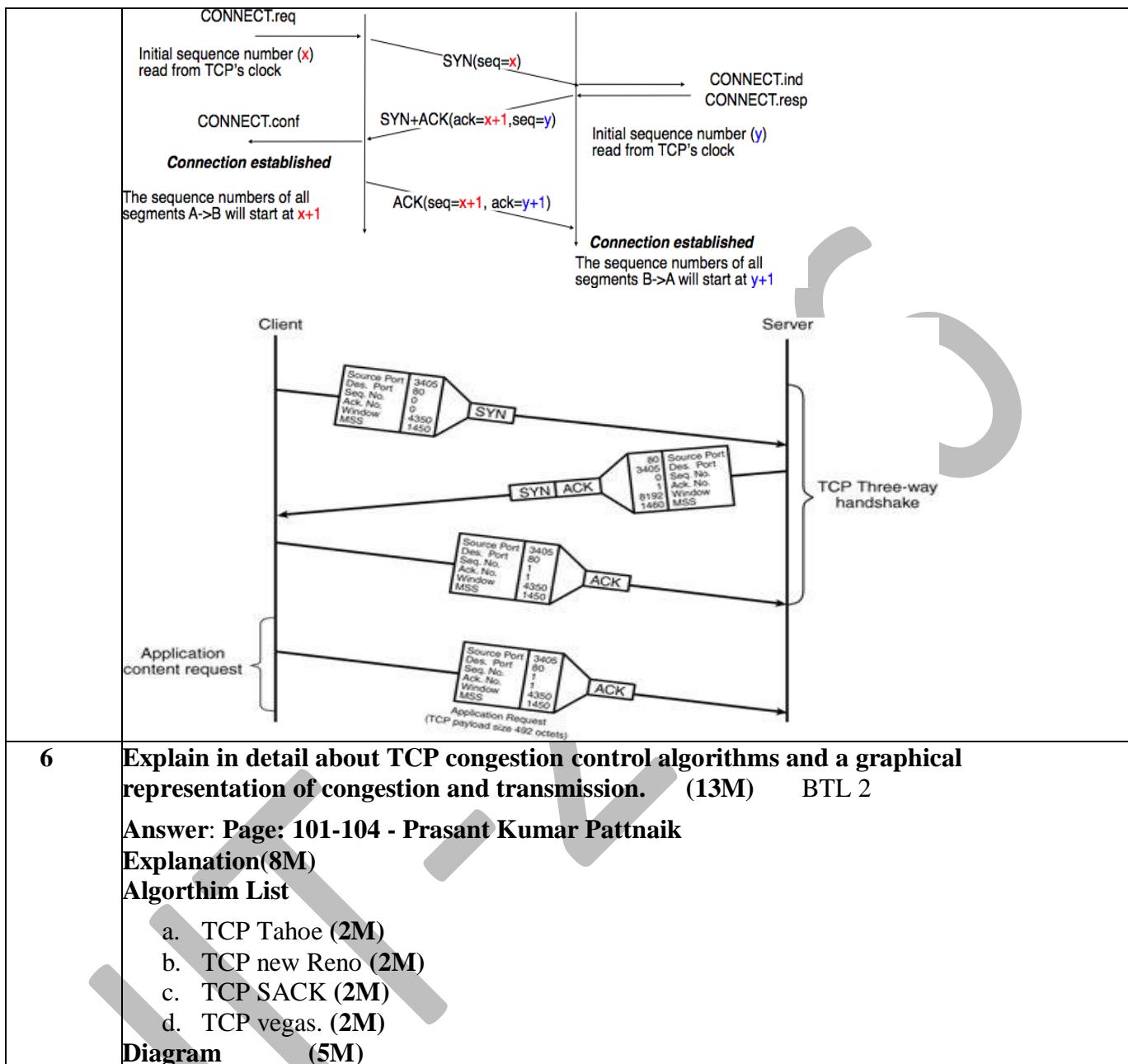
PART * B

1	<p>(i) Discuss improvement in TCP for traditional networks. (6M) BTL 2</p> <p>Answer: Page:88-91 - Prasant Kumar Pattnaik</p> <p>Explanation (4M)</p> <p>Traditional Networks:</p> <ul style="list-style-type: none"> • In the wired networks - packet losses are primarily attributable to congestions - built-up the networks - reduce congestion - TCP invokes congestion control mechanisms. • Congestion control - primarily achieved by reducing transmission window - which in turn results in slower data transfer. <p>Diagram (2M)</p> <p>(ii) Describe Mechanism for TCP Improvement. (7M) (MAY/JUNE 2016) BTL 2</p> <p>Explanation (4M)</p> <p>Slow Start:</p> <ul style="list-style-type: none"> • TCP session is started - starting transmission at a fixed transmission window size - transmission is started at the lowest window size - then doubled after each successful transmission. <p>Congestion avoidance:</p> <ul style="list-style-type: none"> • It starts where slow start stops -once the congestion window reaches the congestion threshold level. <p>Fast retransmit/ fast recovery</p>
---	---

	<ul style="list-style-type: none"> A sender initiates a timer after transmitting a packet - sets timeout value (RTO) - RTO is calculated on RTT. <p>Diagram (3M)</p>	
2	<p>Explain the GSM system architecture with a neat diagram. (13M) (Nov/dec 2011/2015) BTL 2</p> <p>Answer: Page: 40-43 - Prasant Kumar Pattnaik</p> <p>GSM ARCHITECTURE</p> <p>The architecture of GSM comes in hierarchy, consisting of many entities, interfaces and subsystems.</p> <p>The GSM system consist of three subsystems namely,</p> <ul style="list-style-type: none"> The Radio Subsystems(RSS) Network and Switching Subsystems(NSS) Operation Subsystem(OSS) <p>Diagram(5M)</p> <p>Explanation(8M)</p> <p>The components of RSS</p> <ul style="list-style-type: none"> Mobile station Base Transceiver Station Base Station Subsystem Base Station Controller <p>NETWORK AND SWITCHING SUBSYSTEM:</p> <p>Network And Switching Subsystem is the heart of GSM.</p> <p>Functions:</p>	(Nov/dec 2011/2015) BTL 2

	<ul style="list-style-type: none"> • Connects wireless network with standard public network • Performs handover between different BSS • Localization (to locate the mobile station) • Charging, accounting and roaming of users. <p>Functions</p> <ul style="list-style-type: none"> • Traffic monitoring • Status reporting of network entities. • Security management
3	<p>Describe the services provided by GSM with a neat diagram. (13M) (Nov/Dec 2014, 2016) BTL 2</p> <p>Answer: Page: 40-42- Prasant Kumar Pattnaik</p> <p>Definition(2M)</p> <p>GSM: Mobile Services</p> <p>GSM offers several types of connections voice connections, data connections, short message service, multi-service options (combination of basic services).</p> <p>Explanation(6M)</p> <p>Three service domains</p> <ul style="list-style-type: none"> • Bearer Services (2M) • Telematic Services (2M) • Supplementary Services (2M) <p>Bearer Services – interface to the physical medium (transparent for example in the case of voice or non transparent for data services)</p> <p>Telematic Services – services provided by the system to the end user (e.g., voice, SMS, fax, etc.)</p> <p>Supplementary Services – associated with the tele services: call forwarding, redirection, etc.</p> <p>Diagram(5M)</p> <pre> graph LR MS[MS] --- TE1[TE] MS --- MT[MT] TE1 --- R[] MT --- S[] TE1 --- Um[U_m] GSM[GSM-PLMN] --- GSM GSM --- TN[transit network PSTN, ISDN] TN --- SDN[source/destination network] SDN --- TE2[TE] SDN --- U[U, S, R] subgraph tele_services [tele services] TE1 MT GSM TN SDN TE2 end </pre>
4	<p>Explain in detail about Mobile IP with a neat sketch. (13M) BTL 1</p> <p>Answer: Page: 73-77 - Prasant Kumar Pattnaik</p> <p>Definition (2M)</p> <p>Mobile IP (or MIP) is an Internet Engineering Task Force (IETF) standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address.</p>

	<p>Explanation (6M)</p> <ul style="list-style-type: none"> • Routing • Specific routes to end-systems • Changing the IP-address • Transparency • Compatibility • Efficiency and scalability • Home Agent (HA) • Foreign Agent (FA) • •  <p>Diagram (5M)</p> <table border="1"> <thead> <tr> <th>ver.</th><th>IHL</th><th>DS (TOS)</th><th colspan="2">length</th></tr> </thead> <tbody> <tr> <td colspan="2">IP identification</td><td>flags</td><td colspan="2">fragment offset</td></tr> <tr> <td>TTL</td><td colspan="2" rowspan="3">IP-in-IP</td><td colspan="2" rowspan="3">IP checksum</td></tr> <tr> <td colspan="5">IP address of HA</td></tr> <tr> <td colspan="5">Care-of address of COA</td></tr> <tr> <th>ver.</th><th>IHL</th><th>DS (TOS)</th><th colspan="2">length</th></tr> <tr> <td colspan="2">IP identification</td><td>flags</td><td colspan="2">fragment offset</td></tr> <tr> <td>TTL</td><td colspan="2" rowspan="4">lay. 4 prot.</td><td colspan="2" rowspan="4">IP checksum</td></tr> <tr> <td colspan="5">IP address of CN</td></tr> <tr> <td colspan="5">IP address of MN</td></tr> <tr> <td colspan="5">TCP/UDP/ ... payload</td></tr> </tbody> </table>	ver.	IHL	DS (TOS)	length		IP identification		flags	fragment offset		TTL	IP-in-IP		IP checksum		IP address of HA					Care-of address of COA					ver.	IHL	DS (TOS)	length		IP identification		flags	fragment offset		TTL	lay. 4 prot.		IP checksum		IP address of CN					IP address of MN					TCP/UDP/ ... payload				
ver.	IHL	DS (TOS)	length																																																					
IP identification		flags	fragment offset																																																					
TTL	IP-in-IP		IP checksum																																																					
IP address of HA																																																								
Care-of address of COA																																																								
ver.	IHL	DS (TOS)	length																																																					
IP identification		flags	fragment offset																																																					
TTL	lay. 4 prot.		IP checksum																																																					
IP address of CN																																																								
IP address of MN																																																								
TCP/UDP/ ... payload																																																								
5	<p>Elaborate TCP operation in detail. Construct the connection transfer of packets from source to destination with a neat diagram (13M) BTL 6</p> <p>Answer: Page: 92-95 - Prasant Kumar Pattnaik</p> <p>List (3M)</p> <p>A TCP Connection Connection Establishment Three-Way Handshaking: Data Transfer Connection Termination</p> <p>Diagram (10M)</p>																																																							



6 Explain in detail about TCP congestion control algorithms and a graphical representation of congestion and transmission. (13M) BTL 2

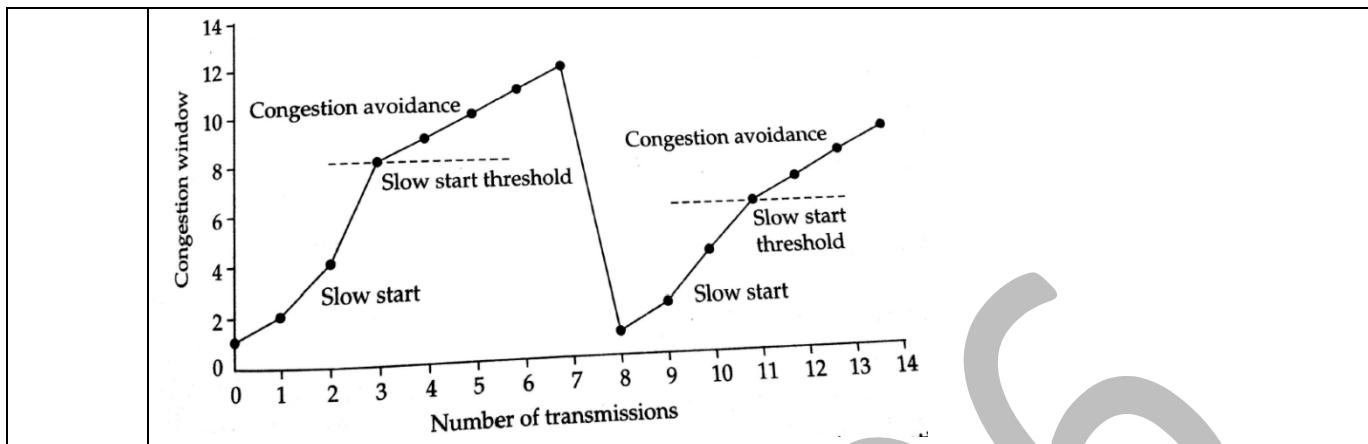
Answer: Page: 101-104 - Prasant Kumar Pattnaik

Explanation(8M)

Algoirthm List

- a. TCP Tahoe (2M)
- b. TCP new Reno (2M)
- c. TCP SACK (2M)
- d. TCP vegas. (2M)

Diagram (5M)

**PART-C**

- 1** Illustrate the comparison of various TCP advantages and disadvantages in Wireless Networking. (15M) (Nov/Dec 2016) BTL 2

Answer: Page: 99-110 - Prasant Kumar Patnaik

Comparison(15M)

TCP approach	Mechanism Used	Merits	Demerits
Indirect TCP(I-TCP)	TCP Segments the connection into two	- Simple - Isolation of wire and wireless links is possible	- Loss of the TCP semantics - Security Problem
Snooping TCP(S-TCP)	Snooping of data and acknowledgements	- Transparency - MCA interaction	- Inadequate isolation of the wireless links - Security problem
Mobile TCP	The segmented TCP connection can choke the sender through window sizes	- End-to-end segment maintained - Handles frequent disconnections	- poor isolation - Wireless links. - Security problem
Fast retransmission Fast recovery	It avoids slow-start after any roaming	- Simple - More efficient	- Not transparent - Mixed Layers
Freeze- TCP	It freezes the TCP later it resumes the TCP after reconnection.	- Works even when there are long interruptions	- Changes in TCP. - MAC dependent

- 2** Explain the architecture of GPRS and transmission Protocol of GPRS (NOV/DEC 2014) BTL 2 (13M)

Answer: Page:44-45 - Prasant Kumar Patnaik

Definition(2M)

GPRS stands for General Packet Radio Services. This mechanism is flexible and powerful. This method provides packet mode for data transfer for small volumes of data, to increase the data transfer.

Explanation(8M)

	<p>Expectations:</p> <ul style="list-style-type: none"> • Should use the existing network resources efficiently. • Should provide the selection of QoS parameters. • Should provide unicast, broadcast, multicast services. <p>Types of packet transfer services.</p> <ul style="list-style-type: none"> • Point to point packet transfer services. • Point to multipoint services. <p>Diagram(3M)</p>
3	<p>Explain System Architecture of UMTS with a neat sketch. (15M) (NOV/DEC 2016) BTL 1</p> <p>Explanation (10M) Answer: Page: 46-48 - Prasant Kumar Pattnaik</p> <p>The UTRA network (UTRAN) handles cell level mobility and comprises several radio network subsystems (RNS).</p> <p>Diagram (5M)</p> <p>The diagram illustrates the UMTS system architecture. It is divided into two main parts: the Core Network (CN) and the UTRAN (Radio Network Subsystem). Core Network (CN): This part includes the GMSC (Gateway Mobile Switching Center), VLR (Visitor Location Register), HLR (Home Location Register), and GGSN (Gateway GPRS Support Node). The GMSC is connected to other networks. The VLR and HLR are connected to each other. The GGSN is connected to an IP network. UTRAN: This part includes Radio Network Controllers (RNC) and Base Stations (BS). The RNCs are connected to the CN via interfaces I_ur and I_ub. The BSs are connected to the RNCs via interfaces I_ub. The BSs also interface with User Equipment (UE) through USIM / Mobile Equipment (ME). The diagram shows two hexagonal cells representing the coverage area of the BSs.</p>

Subject Code: CS 8601
Subject Name: Mobile Computing

Year / Sem : III / 6
Subject Handler: Ms. Suganya M

UNIT III MOBILE TELECOMMUNICATION SYSTEM

Mobile IP – DHCP – AdHoc– Proactive protocol-DSDV, Reactive Routing Protocols – DSR, AODV , Hybrid routing –ZRP, Multicast Routing- ODMRP, Vehicular Ad Hoc networks (VANET) –MANET Vs VANET – Security.

PART * A

1	Define Adhoc network. BTL 1 It is a local area network (LAN) that is built spontaneously as device connects. Instead of relying on a base station to co ordinate the flow of messages to each node in the network, the individual network nodes forward packet to and from each other.
2	Define MANET. BTL 1 Mobile Adhoc network without the support of any form of fixed infrastructure such as a base station or an access point. They are set up among the hand held devices of mobile users.
3	Define VANET. BTL 1 The Vehicular Ad-Hoc Network, or VANET , is a technology that uses moves cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range
4	Which DSDV adds two components to the distance vector algorithm? BTL 1 Sequence Number and Damping
5	Which DSR adds two components to the distance vector algorithm? BTL 1 <ul style="list-style-type: none"> • Route Discovery • Route Maintenance
6	Distinguish between MANET and VANET (NOV/DEC 2016) BTL 2 The main difference between VANET and MANET network is production cost, the VANET production cost is costly when we compare with MANET. The network topology of VANET is frequent, fast, mobility is high because of speed of cars, and other hand the MANET is sluggish and slow. The bandwidth in VANET is higher compare to Mobile Ad-hoc networks. The nodes are moving randomly in MANET but in VANET the nodes are moving regularly.
7	What is the key difference between MANET and other wireless networks? BTL 1 <ul style="list-style-type: none"> • No Fixed Routing/Forwarding Infrastructure • Untrusted environment

	<ul style="list-style-type: none"> • No PKI and Online security mechanism.
8	<p>List the characteristics of MANETs. (MAY/JUNE 2016) BTL 2</p> <ul style="list-style-type: none"> • Lack of fixed infrastructure • Dynamic Topologies • Bandwidth constrained, variable capacity links
9	<p>Why is Routing in MANET a complex task? BTL 1</p> <p>It is difficult to have a global node identifier assigned to every node. In a nutshell, the topology of a network Change dynamically as nodes move way or fail.</p>
10	<p>What is mesh based protocol? BTL 1</p> <p>It establishes a mesh of paths that connects the source and destinations. These are most resilient to link failures as well as to node mobility.</p>
11	<p>What are the characteristics of secure Ad hoc networks? BTL 1</p> <ul style="list-style-type: none"> • Availability • Confidentiality • Integrity • Authentication
12	<p>What are the security vulnerabilities of using adhoc network? BTL 1</p> <ul style="list-style-type: none"> • Lack of physical boundary • Low power RF transmissions • Limited computational capabilities • Limited Power supply
13	<p>What is the difference between AODV and standard distance vector algorithm? BTL 1</p> <p>AODV is capable of both unicast and multicast routing. It is a reactive routing protocol, meaning that it establishes a route to a destination only on demand. Routers use distance vector based routing protocols to periodically advertise the routes in their routing tables. Routing information exchanged between typical distance vector based routers is unsynchronized and unacknowledged.</p>
14	<p>What are the features of MANET routing Protocol? BTL 1</p> <ul style="list-style-type: none"> • Capable to identify network topology after changes due to mobility • Topology Maintenance • Scheduling of packet transmission and channel assignment
15	<p>List example of ON – Demand routing protocol. BTL 2</p>

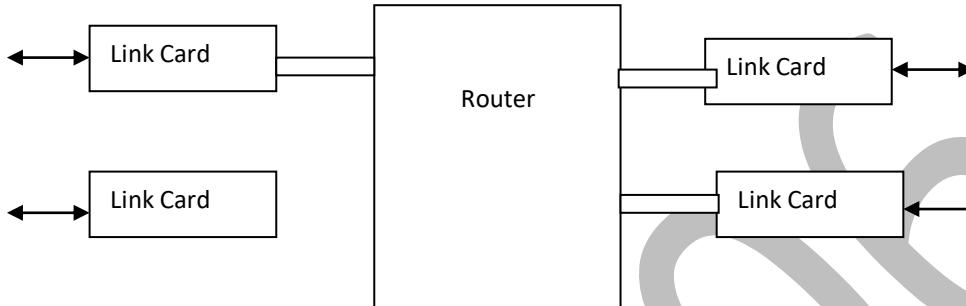
	<ul style="list-style-type: none"> • Dynamic Source Routing (DSR) • Adhoc On-demand distance vector routing (AODV)
16	<p>What are the types of communications in a network? BTL 1</p> <p>The types of communication are:</p> <ul style="list-style-type: none"> • Unicast • Multicast • Broadcast
17	<p>State the reason for topology changes. BTL 3</p> <ul style="list-style-type: none"> • The failure of a mobile node due to battery exhaustion, normal failure, or failure due to adverse environmental condition. • Link disconnections may occur due to noise and changes in signal propagation conditions
18	<p>Define DSR routing. BTL 1</p> <ul style="list-style-type: none"> • The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. • DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. • It is a reactive protocol and all aspects of the protocol operate entirely on-demand basis. • It works on the concept of source routing.
19	<p>Define ODMR. BTL 1</p> <p>On-Demand Multicast routing protocol is a mesh architecture protocol, i.e., it has multiple paths from the sender to the receivers and uses a forwarding group concept.</p> <p>It applies on-demand procedures to dynamically build route and maintain multicast group membership. By maintaining a mesh instead of a tree, the drawbacks of multicast trees in ad hoc networks like frequent tree reconfiguration and non-shortest path in a shared tree are avoided</p>
20	<p>What are the passive and active attacks in MANET? BTL 1</p> <p>Passive: Snooping, eavesdropping, Traffic analysis, Monitoring</p> <p>Active: Wormhole, black hole, resource consumption, routing attacks</p>
21	<p>Define the term 'CGSR'. BTL 4</p> <p>Cluster-Head Gateway Switch Routing Protocol The Cluster-Head Gateway Switch Routing (CGSR) protocol is a table-driven routing protocol. In a clustering system, each predefined number of nodes are formed into a cluster controlled by a cluster head, which is assigned using a distributed clustering algorithm</p>
22	<p>What is the concept of RTT. BTL 3 (NOV/DEC 2016)</p>

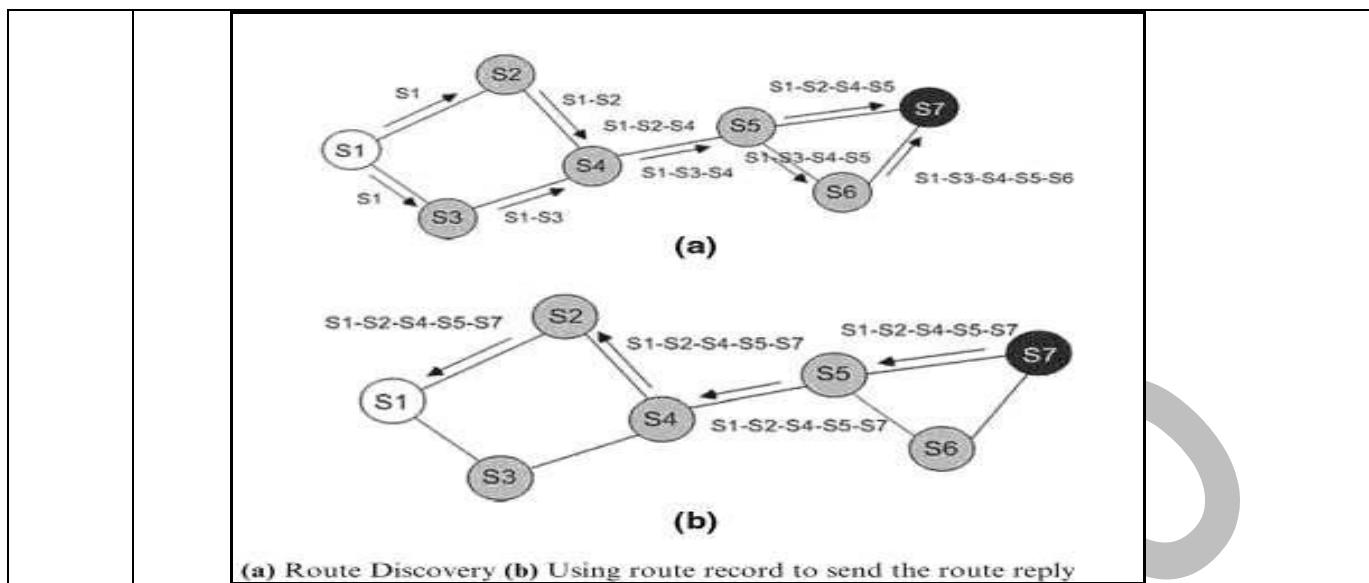
	RTT refers to technology that allows a user to receive data during the actual time that a physical process occurs, known as real time. Real time is measured in milliseconds or microseconds.
23	Distinguish proactive and reactive protocols. BTL 2 (APR/MAY 2017) Reactive and Proactive Protocols are the routing protocols that are used in mobile Ad hoc networks to send data from the host to the destination. A packet data is sent from source to destination in an Ad hoc network through multiple nodes that are mobile.
24	What is multicast routing protocol? BTL 1 A multicast routing protocol is one type of service provider that functions as a client within the framework of the router architecture. A multicast routing protocol manages group membership and controls the path that multicast data takes over the network. Examples of multicast routing protocols include: Protocol Independent Multicast (PIM), Multicast Open Shortest Path First (MOSPF), and Distance Vector Multicast Routing Protocol (DVMRP).
25	List the disadvantage of DSDV. BTL 2 A limitation of DSDV is that it provides only one route for a source/destination pair.

PART * B

Q.No	Questions
1	<p>(i)List the characteristics of Mobile Ad hoc Networks (MANETs) (8M) BTL 2 Answer: Page:130-133 - Prasant Kumar Pattnaik</p> <p>Explanation(8M)</p> <ul style="list-style-type: none"> • Lack of fixed infrastructure • Dynamic Topologies • Bandwidth constrained, variable capacity links • Energy constrained Operation • Increased Vulnerability <p>(ii)explain the design issues of MANET. (5M) BTL 2</p> <p>Explanation(5M)</p> <ul style="list-style-type: none"> • Network Size and node density • Connectivity • Network topology • User traffic • Operational environment • Energy Constraints

<p>2</p> <p>What is VANET? Explain its usage with a neat diagram</p> <p>BTL 2</p> <p>Answer: Page: 152-153 - Prasant Kumar Pattnaik</p> <p>Explanation(8M)</p> <ul style="list-style-type: none"> • Vehicular ad hoc networks (VANET) - special type of MANET - moving automobiles form nodes of the network. • Introduced for vehicles of police - fire brigades and ambulances - safe travelling on road. <p>Diagram(2M)</p> <p>The diagram shows a road with two cars. One car is connected to a roadside unit (RSU) via a dashed line labeled 'V2I'. Two solid lines labeled 'V2V' connect the two cars. The RSU is represented by a cone with a small circle above it, and the cars are represented by dark shapes with small circles above them.</p> <p>Uses of VANET (3M)</p> <p>A VANET can help disseminate geographical information to driver as he continues to drive. For example, the driver would be notified of the nearby food malls or petrol refilling stations, map display.</p>	<p>(13M) (APR/MAY 2017)</p>
<p>3</p> <p>Describe in detail about routing protocols for traditional networks.</p> <p>BTL 2</p> <p>Answer: Page:143-145 - Prasant Kumar Pattnaik</p> <p>Explanation(10M)</p> <p>Link State Protocols (LSP)</p> <p>Link denotes the connection of one router to its neighboring router. A neighbor of a router is one with which it can directly communicate without taking a help from any of the intervening routers.</p> <p>Characteristics</p> <p>Router connection to other routers by network interfaces</p> <ul style="list-style-type: none"> • The identity of the router originating the message. 	<p>(13M) (NOV/DEC 2016)</p>

	<ul style="list-style-type: none"> The identities of all its neighbors The delay along various links to its neighbors. A unique sequence number, which is formed by increasing the count every time the router forms a new link state advertisement  <p>Diagram(3M)</p>
4	<p>Describe in detail about Manet routing protocols. (MAY/JUNE 2016) (13M) BTL 2 Answer: Page: 146-150 - Prasant Kumar Pattnaik</p> <p>Explanation(12M)</p> <p>Destination Sequenced Distance vector routing protocol (4M)</p> <ul style="list-style-type: none"> Table –driven approach to packet routing - extends bellman- ford routing algorithm- improvement made here is avoidance of routing loops through - use of a number sequencing scheme. <p>Steps in operation of DSDV (4M)</p> <ol style="list-style-type: none"> Each router (node) in the network collects route information.. After gathering information - node determines shortest path to destinations based on - gathered information. Based on gathered information - a new routing table is generated. Router broadcasts table to its neighbours- on receipt by neighbours - neighbors nodes recomputed - respective routing tables. Process continues - routing information becomes stable. <p>Dynamic Source routing protocol (DSR) (4M)</p> <ul style="list-style-type: none"> Dynamic Source Routing protocol (DSR) - simple and efficient routing protocol designed specifically - use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows network - completely self-organizing and self-configuring

**Diagram(1M)**

5 What is Routing? State the difference between wired and wireless routing. (13M)
(Nov/Dec 2015) BTL1

Answer: Page:138-139 - Prasant Kumar Pattnaik

Definition(2M)

- **Routing** is the means of discovering paths in computer networks along which information (split up into packets) can be sent.
- Circuit-based networks, such as the voice telephone network, also perform routing, to find paths for telephone calls through the network fabric.

Explanation(6M)

- Routing - directed by **routing tables**, which maintain a record - best routes to various network locations - keep up with packet arrival rate.
- Small networks may involve hand configuration.

Routing Metrics(3M)

- Hop-Count- It is related to the total number of hops between two nodes.
- Queuing delay- This corresponds to the load of the link ie. the traffic going on in the line.

For wireless networks the metrics are

- Hop-Count
- RTT(Round Trip Time)
- Packet Pair
- ETx (Expected Transmission Count)

Hop-Count:**Advantages (2M)**

- Easy to evaluate
- Simple
- Little Overhead

Shortcomings

	<p>It does not consider</p> <ul style="list-style-type: none"> • Transmit rate • Load • Interference <p>Packet Loss Rate</p>
6	<p>What are reactive and proactive protocols? Specify its advantages and disadvantages. (13M) (NOV/DEC 2016) BTL 1</p> <p>Answer: Page: 139-141- Prasant Kumar Patnaik</p> <p>Explanation(8M)</p> <p>Proactive (Table-driven) protocols:</p> <p>Table-driven routing protocol - each node in routing table maintains information about routes - every other node in network.</p> <p>Tables are periodically updated in face of random network topology changes.</p> <p>Example of Proactive - destination Sequenced Distance Vector (DSDV)</p> <p>Example of on-demand routing protocols are:</p> <ul style="list-style-type: none"> • Dynamic source routing(DSR) • Adhoc on- demand distance vector routing (AODV) <p>Diagram(5M)</p>
	PART*C
1	<p>i) Discuss the characteristics of MANET. (8M) (MAY/JUNE 2016) BTL 3 (ii) Summarize the applications of MANET. (7M) BTL 2</p> <p>Answer: Page: 151-153 - Prasant Kumar Patnaik</p> <p>Characteristics: (8M)</p> <ul style="list-style-type: none"> • Lack of fixed infrastructure • Dynamic Topologies • Bandwidth constrained, variable capacity links • Energy constrained Operation • Increased Vulnerability <p>Application: (7M)</p> <ul style="list-style-type: none"> • Tree – based protocol (4M) • Mesh based Protocol: (3M) <p>Example of this category protocol: On-demand Multicast routing protocol(ODMRP)</p>
2	<p>Explain the major types of security attacks in a mobile ad hoc network. (15M)</p> <p>Answer: Page: 129-131- Prasant Kumar Patnaik</p>

	<p>Explanation (15M)</p> <ul style="list-style-type: none"> • DATA traffic attacks • CONTROL traffic attacks • Black-Hole • Cooperative Black-Hole • Gray-Hole • Worm-Hole • HELLO Flood • Bogus Registration • Jellyfish • Man in Middle • Rushing • Cache Poisoning • Blackmail • Cooperative Blackmail • Sybil
3	<p>Explain Distance Vector (DV) protocols in detail with a neat diagram (15M) BTL 3</p> <p>Answer: Page: 141-143 - Prasant Kumar Pattnaik</p> <p>Definition(2M)</p> <p>Routers using distance vector protocol do not have knowledge of the entire path that a packet would take to reach its destination</p> <ul style="list-style-type: none"> • Direction in which a packet should be forwarded. • Its own distance from the destination. <p>Explanation(8M)</p> <p>The Two popular distance vector protocols are:</p> <p>(a) RIP (Routing Information protocol)- It uses hop count of the destination..It supports cross platform distance vector routing</p> <p>(b) IGRP (Interior gateway Routing protocol)- It takes into an account the other</p>

	<p>information such as node delay and available bandwidth. It supports Cisco Systems proprietary distance vector.</p> <p>(c) Ciscos Enhanced IGRP (EIGRP), it doesn't require transmitting updates periodically. Further, the updates are not broadcast and do not contain the full route table.</p> <p>Diagram (5M)</p>
--	--



Subject Code: CS 8601
Subject Name: Mobile Computing

Year / Sem : III / 6
Subject Handler: Ms. Suganya M

UNIT IV MOBILE AD-HOC NETWORKS		
Mobile TCP– WAP – Architecture – WDP – WTLS – WTP –WSP – WAE – WTA Architecture – WML		
PART * A		
1	Define Mobile TCP.	BTL 1 M-TCP (mobile TCP) approach has the same goals as I-TCP and snooping TCP: to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems.
2	List the advantages of Mobile TCP.	BTL 1 ➤ It maintains the TCP end-to-end semantics. The SH does not send any ACK itself but forwards the ACKs from the MH. ➤ If the MH is disconnected, it avoids useless retransmissions, slow starts or breaking connections by simply shrinking the sender's window to 0. Since it does not buffer data in the SH as I-TCP does, it is not necessary to forward buffers to a new SH. Lost packets will be automatically retransmitted to the new SH.
3	List the disadvantages of Mobile TCP.	BTL 1 ➤ As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption. ➤ A modified TCP on the wireless link not only requires modifications to the MH protocol software but also new network elements like the bandwidth manager.
4	Comment WAP transaction layer with its wireless transaction protocol.	BTL 1 The WAP transaction layer with its wireless transaction protocol (WTP) offers a lightweight transaction service at the transaction SAP (TR-SAP). This service efficiently provides reliable or unreliable requests and asynchronous transactions as explained in the above section. Tightly coupled to this layer is the next higher layer, if used for connection-oriented service
5	State Error code.	BTL 1 An error code (EC) is returned indicating the reason for the error to the higher layer. WDP is not allowed to use this primitive to indicate problems with the bearer service.
6	What is WTLS?	BTL 1 WTLS can provide different levels of security (for privacy, data integrity, and authentication) and has been optimized for low bandwidth, high-delay bearer networks.
7	How WTP achieves reliability?	BTL 1 WTP achieves reliability using duplicate removal, retransmission, acknowledgements and unique transaction identifiers. No WTP-class requires any connection set-up or tear-down phase. This avoids unnecessary overhead on the communication link.

8	What are the features of WTP? A special feature of WTP is its ability to provide a user acknowledgement or, alternatively, an automatic acknowledgement by the WTP entity. If user acknowledgement is required, a WTP user has to confirm every message received by a WTP entity.	BTL 1
9	State general features needed for content exchange between cooperating clients and servers. BTL 1 <ul style="list-style-type: none"> ➤ Session management ➤ Capability negotiation ➤ Content encoding 	
10	What are the ideas of Wireless Application Environment? The main idea behind the wireless application environment (WAE) is to create a general-purpose application environment based mainly on existing technologies and philosophies of the world wide web. This environment should allow service providers, software manufacturers, or hardware vendors to integrate their applications so they can reach a wide variety of different wireless platforms in an efficient way.	BTL 1
11	Define WTA. BTL 1 WTA is a collection of telephony specific extensions for call and feature control mechanisms, merging data networks and voice networks.	
12	List the basic features of WML. <ul style="list-style-type: none"> ➤ Text and images ➤ User interaction ➤ Navigation ➤ Context management 	BTL 2
13	What is WAE? Wireless Application Environment, or WAE, provides an architecture for communication between wireless devices and Web servers. ... That Web server responds with an HTML page, which is also sent via HTTP. Because all browsers speak HTTP and both client and server speak the same protocol, they can communicate directly.	BTL 1
14	What are the capabilities not supported by WML? <ul style="list-style-type: none"> ➤ WMLScript offers several capabilities not supported by WML: ➤ Validity check of user input ➤ Access to device facilities ➤ Local user interaction ➤ Extensions to the device software 	BTL 1
15	What are the six libraries in WML Script? <ul style="list-style-type: none"> ➤ Lang 	BTL 1

	<ul style="list-style-type: none"> ➤ Float ➤ String ➤ URL
16	<p>What are the wide range of wireless devices? BTL 1</p> <p>The forum is embracing and extending existing standards and technologies of the internet wherever possible and is creating a framework for the development of contents and applications that scale across a very wide range of wireless bearer networks and wireless device types.</p> <ul style="list-style-type: none"> ➤ Interoperable ➤ Scalable ➤ Efficient
17	<p>State WML Script. BTL 2</p> <p>WMLScript complements to WML and provides a general scripting capability in the WAP architecture (WAP Forum, 2000h). While all WML content is static (after loading on the client)</p>
18	<p>What is Validity Check? BTL 1</p> <p>Validity check of user input, before user input is sent to a server, WMLScript can check the validity and save bandwidth and latency in case of an error. Otherwise, the server has to perform all the checks, which always includes at least one round-trip if problems occur.</p>
19	<p>List DHCP Features. BTL 2</p> <ul style="list-style-type: none"> ➤ DHCP supporting the acquisition of care-of-address for mobile nodes ➤ A DHCP server should located in the subnet of the access point of the mobile note. ➤ DHCP relay should provide forwarding of the Messages. ➤ RFC 3118 specifies authentication for DHCP messages which id needed to protect mobile nodes from malicious DHCP servers.
20	<p>What is Mobile Adhoc Routing? BTL 1</p> <ul style="list-style-type: none"> ➤ In wireless networks using an infrastructure cells have been defines. within a cell the base station can reach all mobile nodes. ➤ In -hoc networks each node must be able to forward data for other nodes. At a certain time t1 the network topology consists of five nodes N1 to N5. ➤ Nodes are connected depending upon the current transmission characteristics between them. In this network N4 can receive N1 over a good link.
21	<p>Define Multicast Routing. BTL 1</p> <p>Multicast IP Routing protocols are used to distribute data (for example, audio/video streaming broadcasts) to multiple recipients. Using multicast, a source can send a single copy of data to a single multicast address, which is then distributed to an entire group of recipients.</p>
22	<p>What is Multicast Group Membership Discovery? BTL 1</p> <p>A protocol is used by receiving hosts to advertise their group membership to a local multicast</p>

	router, enabling them to join and leave multicast groups. The main Multicast Group Membership Discovery protocols are Internet Group Management Protocol (IGMP) for IPv4 and Multicast Listener Discovery (MLD) for IPv6.
23	State Slow Start. BTL 2 TCP's reaction to a missing acknowledgement is quite drastic but it is necessary to get rid of congestion quickly. The behavior shows after the detection of congestion is called Slow start. The sender always calculates a Congestion window for a receiver.
24	Comment Round Trip Time (RIT). BTL 2 This scheme doubles the congestion window every time the acknowledgements come back which takes one Round Trip Time (RIT). This is called the exponential growth of the congestion window in the slow start mechanism.
25	What is Congestion Threshold? BTL 1 It is too dangerous to double the congestion window each time because the steps might become too large. The exponential growth stops at the Congestion Threshold. The congestion window reaches the congestion threshold further increase of the transmission rate is only linear by adding 1 to the congestion window each time the acknowledgements come back.

PART * B

- 1 **What is Mobile TCP? Explain in detail about Mobile TCP. Mention its advantages. (13M)**
BTL 3

Answer: Page:170-173 - Prasant Kumar Pattnaik

The M-TCP (mobile TCP)¹ approach - same goals as I-TCP and snooping TCP: to prevent the sender window from shrinking - if bit errors or disconnection but not congestion cause current problems. M-TCP - improve overall throughput, to lower the delay, to maintain end-to-end semantics of TCP - provide a more efficient handover.

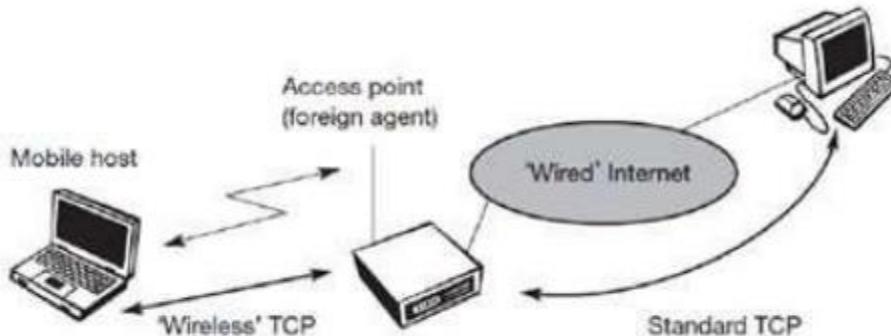
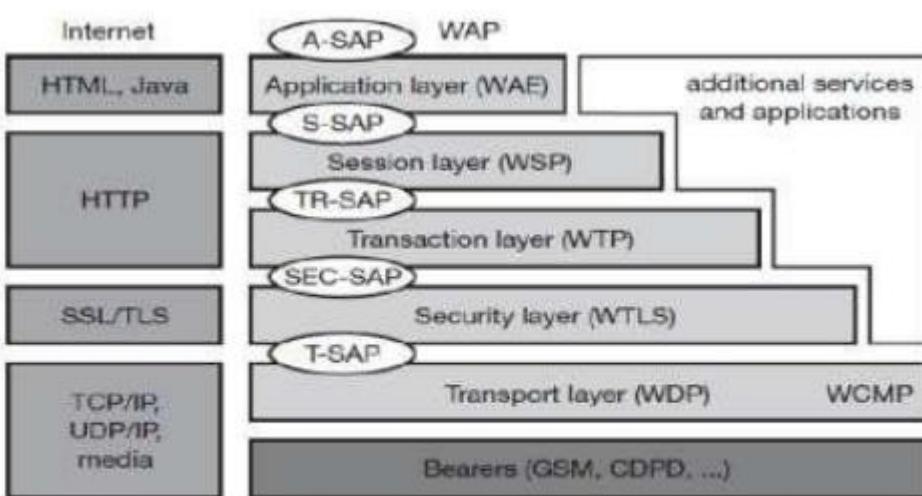


Fig 4.1 Indirect TCP segments a TCP Connection into two parts

Advantages:

1. As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to

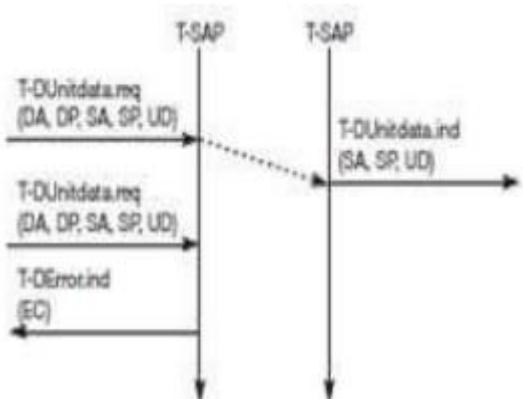
	<p>bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption.</p> <p>2. A modified TCP on the wireless link not only requires modifications to the MH protocol software but also new network elements like the bandwidth manager.</p>
2	<p>Explain in detail about WAP Architecture with a neat sketch. (13M) BTL 3 Answer: Page:129-131 - Prasant Kumar Pattnaik</p> <p>WAP does not specify bearer services, but uses existing data services and will integrate further services. Examples are message services, such as short message service (SMS) of GSM, circuit-switched data, such as high-speed circuit switched data (HSCSD) in GSM, or packet switched data, such as general packet radio service (GPRS) in GSM.</p> <p>Many other bearers are supported, such as CDPD, IS-136, PHS. No special interface has been specified between the bearer service and the next higher layer, the transport layer with its wireless datagram protocol (WDP) and the additional wireless control message protocol (WCMP), because the adaptation of these protocols are bearer-specific.</p>  <pre> graph TD subgraph Internet [Internet] direction TB I1[HTML., Java] I2[HTTP] I3[SSL/TLS] I4[TCP/IP, UDP/IP, media] end subgraph WAP [WAP] direction TB A[A-SAP] --- AL[Application layer (WAE)] AL --- S[S-SAP] --- SL[Session layer (WSP)] SL --- TR[TR-SAP] --- TL[Transaction layer (WTP)] TL --- SEC[SEC-SAP] --- SL2[Security layer (WTLS)] SL2 --- T[T-SAP] --- TLP[Transport layer (WDP)] TLP --- WC[W CMP] WC --- BE[Bearers (GSM, CDPD, ...)] end I1 --- AL I2 --- AL I3 --- AL I4 --- AL AL --- SL SL --- TL SL --- SEC SEC --- TLP TLP --- BE </pre> <p>Fig 4.1 Components and Interface of WAP Architecture</p>

3**Outline Wireless Datagram Protocol with a neat diagram. (13M) BTL 3****Answer: 111-115 - Prasant Kumar Pattnaik**

The Wireless Datagram Protocol (WDP) operates on top of many different bearer services capable of carrying data. At the T-SAP WDP offers a consistent datagram transport service independent of the underlying bearer.

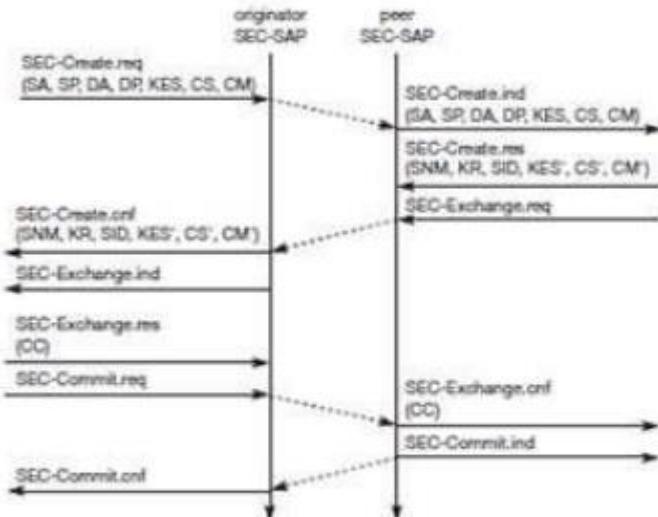
To offer this consistent service, the adaptation needed in the transport layer can differ depending on the services of the bearer. The closer the bearer service is to IP, the smaller the adaptation can be. If the bearer already offers IP services, UDP is used as WDP. WDP offers more or less the same services as UDP.

WDP offers source and destination port numbers used for multiplexing and demultiplexing of data respectively. The service primitive to send a datagram is TDUnitdata.req with the destination address (DA), destination port (DP), Source address (SA), source port (SP), and user data (UD) as mandatory parameters

**Fig 4.3 WAP Service Primitives****4****What is WLTS ? . Brief in detail about WLTS with a outline sketch. (13M) BTL 3****Answer: Page:177-180 - Prasant Kumar Pattnaik**

If requested by an application, a security service, the wireless transport layer security (WTLS), can be integrated into the WAP architecture on top of WDP as specified in (WAP Forum, 2000c).

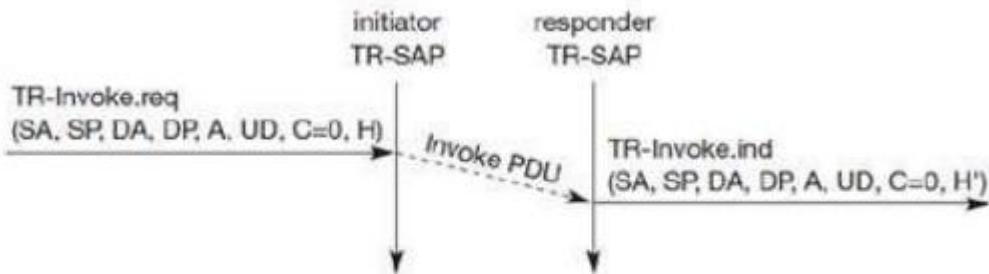
WTLS can provide different levels of security (for privacy, data integrity, and authentication) and has been optimized for low bandwidth, high-delay bearer networks.

**Fig 4.4 WTLS establishing a Secure Session**

- 5** **Describe in detail about Wireless Transaction Protocol. Give a neat sketch. (13M) BTL 3**
Answer: Page:189-193 - Prasant Kumar Pattnaik

WTP has been designed to run on very thin clients, such as mobile phones.

WTP offers several advantages to higher layers, including an improved reliability over datagram services, improved efficiency over connection-oriented services, and support for transaction-oriented services such as web browsing.

**Fig 4.6 Basic Transaction , WTP Class 0**

WTP class 1 Class 1 offers a reliable transaction service but without a result message. Again, the initiator sends an invoke PDU after a TR-Invoke.req from a higher layer.

- 6** **List the features of Wireless Session Protocol. Explain in detail. (13M) BTL 3**
Answer: Page:182-183 - Prasant Kumar Pattnaik

Session management:

WSP introduces sessions that can be established from a client to a server and may be long lived. Sessions can also be released in an orderly manner.

Capability negotiation:

	<p>Clients and servers can agree upon a common level of protocol functionality during session establishment.</p> <p>Content encoding: WSP also defines the efficient binary encoding for the content it transfers. WSP offers content typing and composite objects, as explained for web browsing.</p> <p>HTTP/1.1 functionality: WSP/B supports the functions HTTP/1.1 offers, such as extensible request/reply methods, composite objects, and content type negotiation.</p> <p>Exchange of session headers: Client and server can exchange request/reply headers that remain constant over the lifetime of the session.</p> <p>Push and pull data transfer: Pulling data from a server is the traditional mechanism of the web. This is also supported by WSP/B using the request/response mechanism from HTTP/1.1.</p> <p>Asynchronous requests: Optionally, WSP/B supports a client that can send multiple requests to a server simultaneously.</p>
	PART-C
1	<p>Describe the main idea behind the Wireless Application Environment(WAE). (13M) BTL 3</p> <p>Answer: Page:212-215 - Prasant Kumar Pattnaik</p> <p>The main idea behind the wireless application environment (WAE) is to create a general-purpose application environment based mainly on existing technologies and philosophies of the world wide web.</p> <p>This environment should allow service providers, software manufacturers, or hardware vendors to integrate their applications so they can reach a wide variety of different wireless platforms in an efficient way.</p> <p>HTML, JavaScript, and the handheld device markup language HDML form the basis of the wireless markup language (WML) and the scripting language WML script.</p> <pre> graph LR subgraph OS [Origin servers] WS[Web server] OCS[Other content servers] end G[Gateway] C[Client] WS -- "Response with content" --> G G -- "Encoded response with content" --> C C -- "Encoded push content" --> G G -- "Encoded request" --> C </pre> <p>Fig 4.10 WAE Logical Model</p>

(e.g., the WML browser) and allows network operators to increase accessibility for various special services in their network. A network operator can reach more end-devices using WTA because this is integrated in the wireless application environment (WAE) which handles device-specific characteristics and environments.

WTA extends the basic WAE application model in several ways:

- Content push
- Access to telephony functions
- Repository for event handlers
- Security model

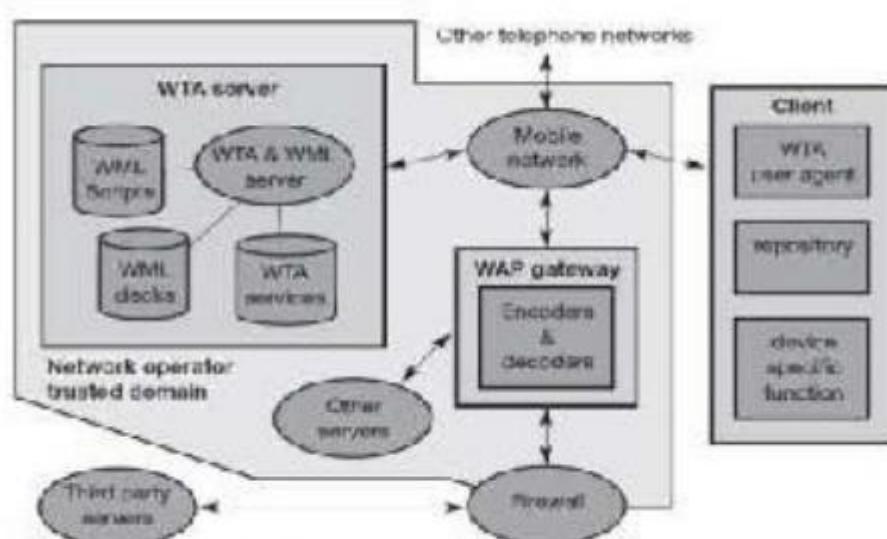


Fig 4.11 WTA Architecture

3

Outline the Wireless Markup Languages with sample code. (13M) BTL 3
Answer: Page:232-233 - Prasant Kumar Pattnaik

The wireless link will always have only a very limited capacity compared to a wire. Current handheld devices have small displays, limited user input facilities, limited memory, and only low performance computational resources. While the bandwidth argument will remain for many years, it currently seems that the gap between mobile and fixed devices regarding processing power is getting narrower.

<?xml version="1.0"?>

```

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

```

```
<wml>

<card id="card_one" title="Simple example"><do type="accept">

<go href="#card_two"/> </do>

<p>

This is a simple first card! <br/>
On the next one you can choose ...

</p>
</card>
```



Subject Code: CS 8601
Subject Name: Mobile Computing

Year / Sem : III / 6
Subject Handler: Ms. Suganya M

UNIT V MOBILE PLATFORMS AND APPLICATIONS	
Mobile Device Operating Systems – Special Constraints & Requirements – Commercial Mobile Operating Systems – Software Development Kit: iOS, Android, BlackBerry, Windows Phone – MCommerce– Structure – Pros & Cons – Mobile Payment System – Security Issues.	
PART * A	
1	What are the two main responsibilities of OS in a mobile handset? BTL 1 Managing Resources Providing different interfaces.
2	What is the advantage of using Microkernel design approach? BTL 1 It minimizes the size of the kernel code. It is easier to port, extend, and maintain operating system code.
3	List the Special constraints of mobile OS. (MAY/JUNE 2016) BTL 2 <ul style="list-style-type: none"> • Limited memory • Limited Screen Size • Miniature keyboard • Limited processing power
4	List the Special service requirements of mobile OS. BTL 2 <ul style="list-style-type: none"> • Support for specific communication protocols • Support for variety of input mechanisms • Support for IDE • Extensive library support
5	What is the advantage of E-commerce? BTL 1 <ul style="list-style-type: none"> • The benefits of using M-Commerce include customer convenience, cost savings and new business opportunities. • From the customer's perspective, M-Commerce provides the flexibility of anytime, anywhere shopping using just a light weighted device.
6	What is the Disadvantage of E-commerce? BTL 1 <ul style="list-style-type: none"> • Mobile devices do not generally offer graphics or processing power of PC. The users are therefore constrained to use small screen and keyboard and low resolution pictures and videos. It may be difficult to perceive the look and feel of many products from online pictures and videos. • The small screens of mobile devices limit the complexity of applications. For

	<p>example, the menu choice, and txt typing capability are severely constrained.</p> <ul style="list-style-type: none"> • No Security.
7	<p>Define mobile payment system. BTL 1 A mobile payment may be defined as any payment instrument where a mobile device is used to initiate, authorize and confirm an exchange of financial value in return of goods and service.”</p>
8	<p>What are the features of SDK? BTL 1</p> <ul style="list-style-type: none"> • They can run the application on the actual android device or a software emulator on the host machine. • This is achieved by using the android Debug Bridge (ADB) available with SDK
9	<p>List out Android application components. BTL 2</p> <ul style="list-style-type: none"> • Activity • Content Providers • Service • Broadcast receivers.
10	<p>What is the advantage of Android? BTL 1</p> <ul style="list-style-type: none"> • It is an Open platform and can be ported on all cell phone. • The android SDK to develop applications is possible on every operating system. • They support robust libraries for media access, communication and data transfer.
11	<p>What is radio frequency identification? BTL 1 RFID tag can be attached to a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from distance that may be several meters away from the reader and beyond the sight of the reader.</p>
12	<p>List the operating system that is available for sensor nodes. BTL 2</p> <ul style="list-style-type: none"> • Tiny OS • Contiki • Lite OS • Mantis
13	<p>Give some applications of M-commerce. BTL 2</p> <ul style="list-style-type: none"> • Advertising

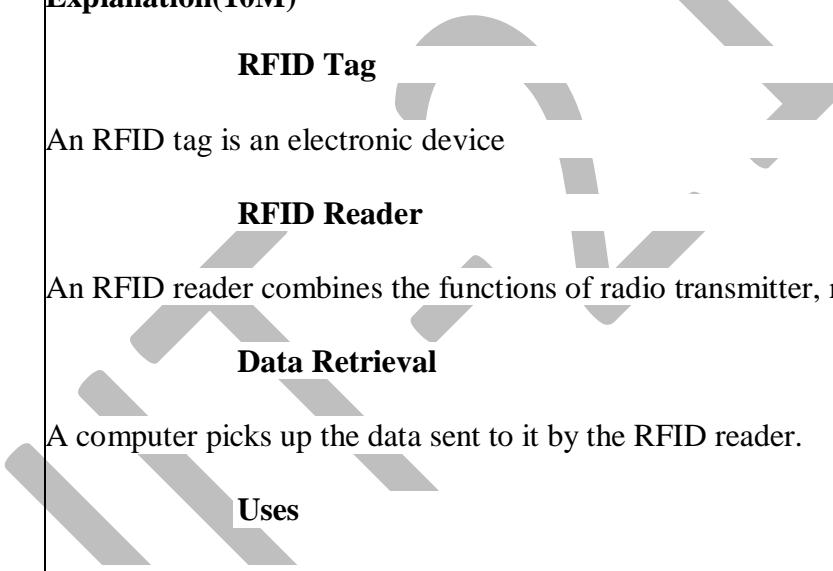
	<ul style="list-style-type: none"> • Mobile ticketing • Loyalty and payment services • Interactive advertisements. 				
14	What are the two popular types of M-payment schemes? BTL 1 <ul style="list-style-type: none"> • Bank account Based • Credit card based • Micro Payment 				
15	What are the features required by a mobile device to enable mobile e commerce? BTL 1 <ul style="list-style-type: none"> • Good Internet Connectivity • Ability to display rich content such as images • Ability to scan bar codes • Ability to read RFID tags 				
16	Define POS. (NOV/DEC 2016) BTL 1 A point of sale (POS) is the place where sales are made. On a macro level, a POS may be a mall, a market or a city. On a micro level, retailers consider a POS to be the area where a customer completes a transaction, such as a checkout counter. It is also known as a point of purchase.				
17	Differentiate E- Commerce and M-Commerce. (NOV/DEC 2016) BTL 2 <table border="1"> <tr> <td>E-Commerce 1. Any kind of commercial transaction that is concluded, over the internet using electronic system is known as e-commerce.</td><td>M-Commerce M-commerce refers to the commercial activities which are transacted with the help of wireless computing devices such as cell phone or laptops.</td></tr> <tr> <td>2. Use of internet is compulsory</td><td>2. Use of internet is not mandatory</td></tr> </table>	E-Commerce 1. Any kind of commercial transaction that is concluded, over the internet using electronic system is known as e-commerce.	M-Commerce M-commerce refers to the commercial activities which are transacted with the help of wireless computing devices such as cell phone or laptops.	2. Use of internet is compulsory	2. Use of internet is not mandatory
E-Commerce 1. Any kind of commercial transaction that is concluded, over the internet using electronic system is known as e-commerce.	M-Commerce M-commerce refers to the commercial activities which are transacted with the help of wireless computing devices such as cell phone or laptops.				
2. Use of internet is compulsory	2. Use of internet is not mandatory				
18	What is Mobile Wallet? BTL 1 A user may have a number of ATM card or credit card. The mobile wallet helps to keep these under the umbrella of a single wallet and can make payments whenever necessary. A few example of mobile wallet are paypal, google wallet, Paytm,etc.				
19	What is mChek? BTL 1 It is a new payment system that links a debit or credit card, or a bank account, to a mobile phone, allowing one to make payments from the mobile phone.				
20	List the disadvantage of M-Commerce? (APR/MAY 2017) BTL 2				

	Mobile device do not generally offer graphics or processing power of a PC. The small screens of mobile devices limit the complexity of applications. Security.
21	What is microkernel operating system? BTL 1 A microkernel is a piece of software or even code that contains the near-minimum amount of functions and features required to implement an operating system.
22	Analyze the features of windows iPhone. BTL 3 <ul style="list-style-type: none"> • Supports iOS 2, iOS 3, iOS 4 and iOS 5 devices • Multi-platform (Java based) product, supported on Linux, Windows and Mac • Fast, powerful search across device including regular expressions • Integrated mapping supports visualisation of geo-tagged information, including google maps searches, photos, and cell-sites and wifi locations observed by the device (the infamous "locationd" data)
23	Describe UIQ interface. BTL 3 An interface is a set of commands or menus through which a user communicates with a program. A command-driven interface is one in which you enter commands. A menu-driven interface is one in which you select command choices from various menus displayed on the screen.
24	What are the elements of Android software stack? (APR/MAY 2017) BTL 1 <ul style="list-style-type: none"> • linux kernel • native libraries (middleware), • Android Runtime • Application Framework • Applications
25	State the drawbacks of Symbian OS. BTL 2 The reason for problems could have been in the software architecture. The basic Symbian OS was pretty well optimized to run on even pretty low power CPUs (uses less processing power than Linux or iOS for the same tasks), but to accomplish this Symbian went its own way with just about everything.
PART * B	
1	i) What are the advantages of M commerce? (7M) BTL 2 Answer: Page: 223-224 - Prasant Kumar Pattnaik Advantages: <ul style="list-style-type: none"> • For business organization- benefits of using M-Commerce - include customer convenience - cost savings - new business opportunities. • customer's perspective - M-Commerce provides - flexibility of anytime, anywhere shopping - light weighted device- customer can save substantial time compared to

	<p>visiting several stores - identifying - right product at the lowest price.</p> <ul style="list-style-type: none"> Mobile devices - highly personalized - providing an additional level of convenience to customers. For example - a repeat order for some items - placed just at touch of a button. Application software downloaded for specific m-commerce vendor - store many customer specific information - help to effortlessly place orders. <p>ii) What are the disadvantages of M commerce? (6M) BTL 2</p> <p>Disadvantages:</p> <ul style="list-style-type: none"> Mobile devices - generally offer graphics or processing power of a PC- users are constrained to use small screen - keyboard and low resolution pictures videos - difficult to perceive - look and feel of many products from online pictures and videos. Small screens of mobile devices limit - complexity of applications. For example- menu choice- text typing capability- severely constrained. Network impose several types of restrictions -for example, the available bandwidth is severely restricted - international calls - SMS be prohibitively expensive. disadvantage is security -unless a customer is extremely careful- may fall prey to various types of frauds - may get billed for items he did not purchase.
2	<p>Explain in detail the structure of Mobile Commerce. (13M) BTL 2</p> <p>Answer: Page: 223-226 - Prasant Kumar Pattnaik</p> <p>Definition(2M)</p> <ul style="list-style-type: none"> In mobile commerce, a content provider implements an application by providing two sets of programs: Client side and server-side. The client side programs run on the micro browsers installed on the users mobile devices. These server side programs, performing database access and computations, reside on the host computer (servers). <p>Explanation(6M)</p> <ul style="list-style-type: none"> Mobile Devices Network Host Computers <p>Major components:</p>

	<ul style="list-style-type: none"> • Web servers. • Database servers • Application Program
	<p>Diagram(5M)</p> <p>3 What are the special constraints of Mobile O/S? Illustrate with examples. (13 M) (NOV/DEC 2016) BTL 3 Answer: Page:230-231 - Prasant Kumar Pattnaik</p> <p>Definition(2M) The operating system for a mobile device needs to function in the presence of many times of constraints which are not present in the traditional computer. As an example of such a constraint, consider the fact that a mobile device is powered by severely limited energy stored in a tiny battery.</p> <p>Explanation(6M)</p> <ul style="list-style-type: none"> • Limited Memory • Limited Screen Size • Miniature Keyboard • Limited Processing Power • Limited Battery Power • Limited and fluctuating bandwidth of the wireless medium • Real Time data streaming
	<p>Diagram(5M)</p> <p>4 Describe in detail about Mobile payment systems. (13M) BTL 3 Answer: Page: 231-232 - Prasant Kumar Pattnaik</p> <p>Definition(2M) Mobile Payment Systems “Mobile payments are a natural evolution of E-payment schemes. A mobile payment may be defined as any payment instrument where a mobile device is used to initiate, authorize and confirm an exchange of financial value in return of goods and service.”</p> <p>Explanation(10M) Mobile Payment Schemes</p> <ul style="list-style-type: none"> • Bank account based • Credit card based

	<ul style="list-style-type: none"> • Micropayment <p>Desirable properties of a Mobile Payment System:</p> <ul style="list-style-type: none"> • Easy to use: • The M-payment request must be easy for the customer to use <p>Mobile Payment solution:</p> <ul style="list-style-type: none"> • SMS based payment: • POS based payment • Bar code based payment • Mobile Wallet <p>Process of Mobile Payment</p> <p>Diagram(1M)</p>
5	<p>Describe in detail about Commercial Mobile operating systems in detail. (13M) (APR/MAY 2017) BTL 3</p> <p>Answer: Page:232-233 - Prasant Kumar Pattnaik</p> <p>Explanation(8M)</p> <ul style="list-style-type: none"> • The Graphic / Window / Event manager (GWE) component handles all input and output • Previous a virtual memory management • Supports security through provision of a cryptographic library. • Application development similar to that in Win32 environment. advantages since many programmers have knowledge of Win 32 based application development <p>Android (5M)</p> <ul style="list-style-type: none"> • Android software stack • Application layer • Application framework • Libraries and runtime • Kernel
6	<p>Discuss the applications of M-Commerce with a neat sketch. (13M) (NOV/DEC 2016) BTL 3</p> <p>Answer: Page: 223-224- Prasant Kumar Pattnaik</p> <p>Explanation(8M)</p> <ul style="list-style-type: none"> • Advertising

	<ul style="list-style-type: none"> • Comparison Shopping • Information about a product • Mobile ticketing • Catalogue Shopping <p>Diagram(5M)</p>
	<p style="text-align: center;">PART * C</p> <p>1 What is RFID? Briefly explain the principle and its working. (15M) (MAY/JUNE 2016) BTL 2 Answer: Page:209-211 - Prasant Kumar Pattnaik</p> <p>Definition (2M) Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically-stored information. The tags contain electronically-stored information.</p> <p>Explanation(10M)</p>  <pre> graph TD Tag[RFID Tag] --> Signals[] Signals --> Reader[RFID Reader] Reader --> Retrieval[Data Retrieval] Retrieval --> Computer[A computer picks up the data sent to it by the RFID reader.] Computer --> Uses[Uses] </pre> <p>An RFID tag is an electronic device</p> <p>An RFID reader combines the functions of radio transmitter, receiver and data interface.</p> <p>A computer picks up the data sent to it by the RFID reader.</p> <p>Many companies use RFID tags to track the flow of goods through warehousing, distribution and retail.</p> <p>Diagram(3M)</p> <p>2 What do you understand by the mobile payment system? Briefly explain an application where mobile payment may be useful. (15M) (NOV/DEC 2016) BTL 2 Answer: Page: 231-235- Prasant Kumar Pattnaik</p> <p>Definition(2M) Mobile Payment Systems: "Mobile payments are a natural evolution of E-payment schemes. A mobile payment</p>

	<p>may be defined as any payment instrument where a mobile device is used to initiate, authorize and confirm an exchange of financial value in return of goods and service.”</p> <p>Explanation(10M)</p> <p>Mobile Payment Schemes</p> <ul style="list-style-type: none"> • Bank account based • Credit card based • Micropayment <p>Desirable properties of a Mobile Payment System:</p> <ul style="list-style-type: none"> • Easy to use: • The M-payment request must be easy for the customer to use <p>Mobile Payment solution:</p> <p>SMS based payment:</p> <p>POS based payment</p> <p>Bar code based payment</p> <p>Mobile Wallet</p> <p>Process of Mobile Payment</p> <p>Diagram(3M)</p>
3	<p>Explain the different mobile payment schemes and security issues. (15M) (MAY/JUNE 2016) BTL 3</p> <p>Answer: Page:234-235 - Prasant Kumar Pattnaik</p> <p>Explanation(10M)</p> <p>Mobile Payment Schemes</p> <ul style="list-style-type: none"> • Bank account based • Credit card based • Micropayment <p>Step 1: Customer places order for goods with the trader.</p> <p>Step 2: The trader securely transfer the order to the selected payment service provider over the Internet.</p> <p>Step 3: The customer authenticates with the payment service provided.</p> <p>Step 4: The transaction detail appropriately and securely routes the transaction authorization request through its payment gateway to the selected customer’s bank.</p> <p>Step 5: The merchant is informed of the payment status.</p> <p>Step 6: For Successful transaction, the customer’s bank transfer the requested amount to the trader’s bank account.</p> <p>Diagram(5M)</p>

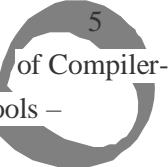
CS8602

COMPILER DESIGN**L TPC****3 0 03****OBJECTIVES:**

- Learn the design principles of a Compiler.
- Learn the various parsing techniques and different levels of translation
- Learn how to optimize and effectively generate machine codes

UNIT I**INTRODUCTION TO COMPILERS**

Translators-Compilation and Interpretation-Language processors -The Phases Errors Encountered in Different Phases-The Grouping of Phases-Compiler Construction Tools – Programming Language basics.

**UNIT II****LEXICAL ANALYSIS**

Need and Role of Lexical Analyzer-Lexical Errors-Expressing Tokens by Regular Expressions- Converting Regular Expression to DFA- Minimization of DFA-Language for Specifying Lexical Analyzers-LEX-Design of Lexical Analyzer for a sample Language.

9

UNIT III**SYNTAX ANALYSIS**

10

Need and Role of the Parser-Context Free Grammars -Top Down Parsing -General Strategies- Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0)Item- Construction of SLR Parsing Table -Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer- YACC-Design of a syntax Analyzer for a Sample Language .

UNIT IV **SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT** 12

Syntax directed Definitions-Construction of Syntax Tree-Bottom-up Evaluation of S-Attribute Definitions- Design of predictive translator – Type Systems-Specification of a simple type checker- Equivalence of Type Expressions-Type Conversions.

RUN-TIME ENVIRONMENT: Source Language Issues-Storage Organization-Storage Allocation- Parameter Passing-Symbol Tables-Dynamic Storage Allocation-Storage Allocation in FORTAN.

UNIT V CODE OPTIMIZATION AND CODE GENERATION 9 Principal Sources of Optimization- DAG- Optimization of Basic Blocks-Global Data Flow Analysis-Efficient Data Flow Algorithms-Issues in Design of a Code Generator – A Simple Code Generator Algorithm.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, students will be able to:

- Design and implement a prototype compiler.
- Apply the various optimization techniques.
- Use the different compiler construction tools.

TEXT BOOKS:

1. Alfred V Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, “Compilers – Principles, Techniques and Tools”, 2nd Edition, Pearson Education, 2007.

REFERENCES:

1. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-based Approach”, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, “Advanced Compiler Design and Implementation”, Morgan Kaufmann Publishers – Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. Charles N. Fischer, Richard. J. LeBlanc, “Crafting a Compiler with C”, Pearson Education, 2008.

Subject Code:CS8602
Subject Name: COMPILER DESIGN

Year/Semester: III /06
Subject Handler: R.Dayana

UNIT I - INTRODUCTION TO COMPLILERS

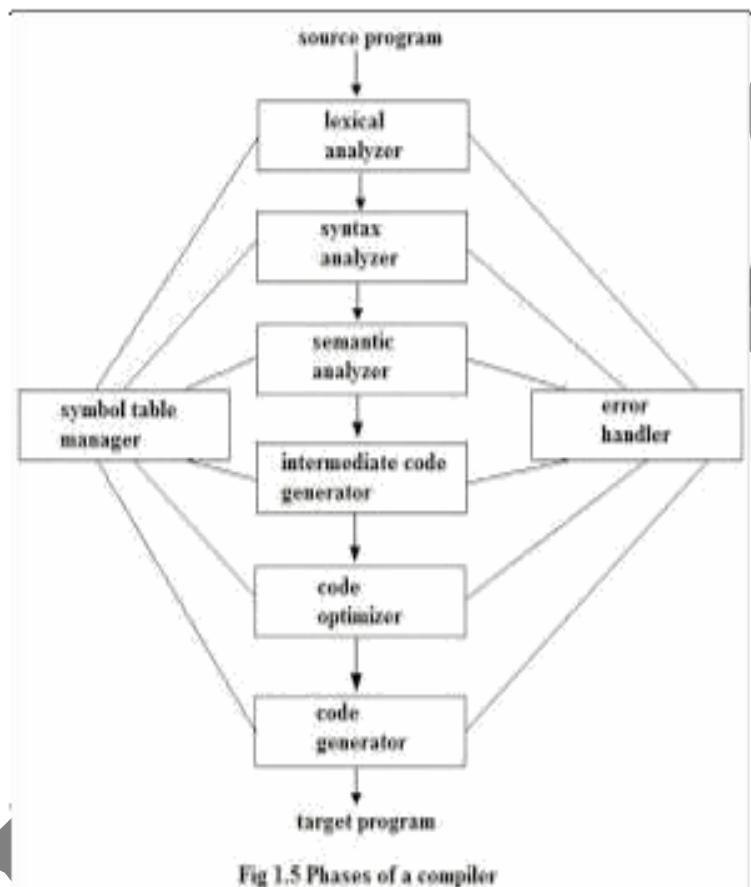
Translators-Compilation and Interpretation-Language processors -The Phases of Compiler-Errors Encountered in Different Phases-The Grouping of Phases-Compiler Construction Tools – Programming Language basics.

PART * A

Q.No.	Questions
	Define is a Complier?
1 A	Complier is a program that reads a program written in one language-the source language-and translates it in to an equivalent program in another language-the target language . As an important part of this translation process, the compiler reports to its user the presence of errors in the source program.
	List the cousins of the compiler? APR/MAY 2017
2	The following are the cousins of i. Preprocessors ii. Assemblers iii. Loaders iv. Link editors.
	Define the two parts of compilation. April/May 2017 May/June 2016
3	Analysis part breaks up the source program into constituent pieces and creates an intermediate representation of the source program. Synthesis part constructs the desired target program from the intermediate representation
	List the various compiler construction tools. Nov /Dec 2016
4	i. Parse generator ii. Scanner generators iii. Syntax-directed translation engines iv. Automatic code generator v. Data flow engines.

	What is a Symbol table? Nov/Dec 2016
5	Symbol table is a data structure containing a record for each identifier, with fields for the attributes of the identifier. The data structure allows us to find the record for each identifier quickly and to store or retrieve data from that record quickly
	Illustrate diagrammatically how a language is Processed. May/June 2016
6	<p>source program</p> <p>↓</p> <p>Preprocessor</p> <p>↓</p> <p>Source program</p> <p>↓</p> <p>Compiler</p> <p>↓</p> <p>Target assembly program</p> <p>↓</p> <p>Assembler</p> <p>↓</p> <p>Relocatable machine code</p> <p>↓</p> <p>Loader/ link editor ←library, relocatable object files</p> <p>↓</p> <p>Absolute machine code</p> 
7	<p>List the phases that constitute the front end of a compiler.</p> <p>The front end consists of those phases or parts of phases that depends primarily on the source language and is largely independent of the target machine. These include</p> <ul style="list-style-type: none"> <input type="checkbox"/> Lexical and Syntactic analysis <input type="checkbox"/> The creation of symbol table

	<input type="checkbox"/> Semantic analysis <input type="checkbox"/> Generation of intermediate code
8	<p>Mention the back-end phases of a compiler.</p> <p>The back end of compiler includes those portions that depend on the target machine and generally those portions do not depend on the source language, just the intermediate language. These include</p> <ul style="list-style-type: none"> <input type="checkbox"/> Code optimization <input type="checkbox"/> Code generation, along with error handling and symbol-table operations <p>List the various phases of a compiler.</p> <p>The following are the various phases of a compiler:</p> <ul style="list-style-type: none"> • Lexical Analyzer • Syntax Analyzer • Semantic Analyzer • Intermediate code generator • Code optimizer • Code generator
9	<p>Define Preprocessor. List its advantages.</p> <p>A preprocessor is one, which produces input to compilers. A source program may be divided into modules stored in separate files. The task of collecting the source program is sometimes entrusted to a distinct program called a preprocessor.</p>
10	<p>The preprocessor may also expand macros into source language statements</p> <p>Identify the functions of Preprocessors.</p> <ul style="list-style-type: none"> Macro processing File inclusion Relational Preprocessors
11	<p>Language extensions</p> <p>Define Structure editor? Summarize the uses of it.</p> <p>A structure editor takes as input a sequence of commands to build a source program. The structure editor not only performs the text creation and modification functions of an ordinary text editor but it also analyzes the program text putting an appropriate hierarchical</p>
12	

	structure on the source program.
	PART * B
	<p>Describe the various phases of Compiler and trace it with the program segment (position=initial + rate * 60). (13M) May/June 2016 Nov/Dec 2016 Apr/May 2017</p> <p>Compiler Definition – (2 Marks)</p> <p>A compiler is a program which translates the <i>source</i> language program into an equivalent program in another language (the <i>target</i> language).</p> <p>Diagram – (3 Marks)</p>  <pre> graph TD SP[source program] --> LA[lexical analyzer] LA --> SA[syntax analyzer] SA --> SA[semantic analyzer] SA --> ICG[intermediate code generator] ICG --> CO[code optimizer] CO --> CG[code generator] CG --> TP[target program] STM[symbol table manager] --> LA STM --> SA STM --> ICG STM --> CO EH[error handler] --> LA EH --> SA EH --> ICG EH --> CG </pre> <p>Fig 1.5 Phases of a compiler</p>

- o Semantic analysis
- o Intermediate code generation
- o Code Optimization
- o Code Generation

Example Problem -(4 Marks)

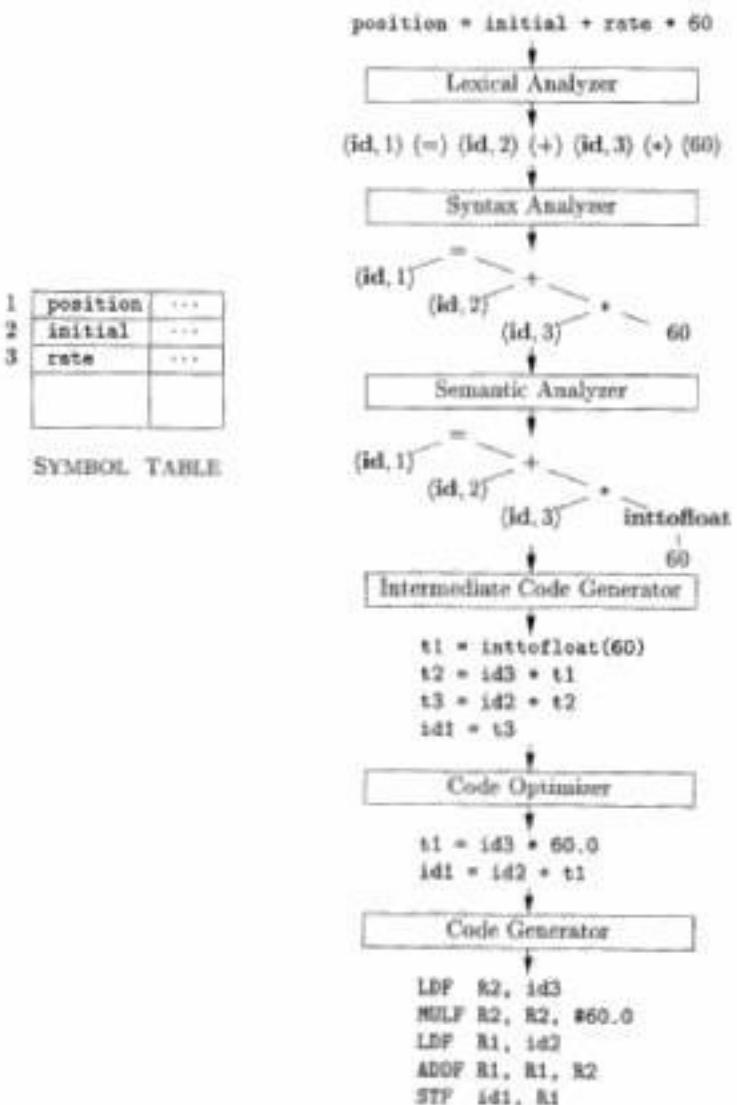


Figure 1.7: Translation of an assignment statement

2

Explain Briefly about Complier Construction Tools (8 Marks)

Compiler Construction Tools (2 Marks)

1. Parser generators
2. Scanner generators
3. Syntax-directed translation engines
4. Automatic code generators
5. Data-flow engines

Explanation – all Tools (6 Marks)

Explain language processing system with neat diagram. (OR) Explain the Cousins of Compiler (8 Marks) May/June 2016 April/May 2017

Cousins of Compiler – (2 Marks)

1. Preprocessors
2. Assemblers
3. Loader
4. Linker

Diagram with Explanation –(6 Marks)

3

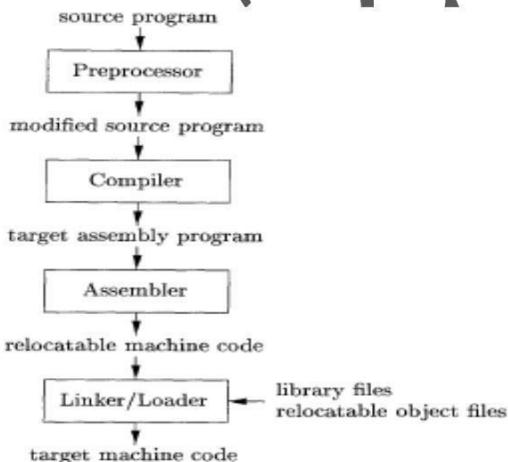


Figure 1.5: A language-processing system

Explain the need for grouping of Phases. (7Marks) Nov/Dec 2016 May/June 2016

Pass -I (4Marks)

Lexical Analysis

Syntax Analysis

Intermediate Code generation

Pass -II (3Marks)

Code generation

4

	Code Optimization
	<p>Explain various errors encountered in different phases of compiler. (6 Marks)</p> <p>May/June 2016</p> <p>During different phases of compiler, all possible errors made by the programmer are detected and they are reported to the user in the form of messages. This process of locating errors and reporting to user is called error handling process.</p> <p>1. Lexical phase Errors: (2 Mark)</p> <p>The lexical phase can errors when the characters remaining in the input do not form any token of the language.</p> <p>Typical errors in this phase are</p> <ul style="list-style-type: none"> • Spelling errors – hence get incorrect tokens. • Exceeding length of identifier or numeric constants. • Appearance of illegal characters. <p>Ex. For Spelling error :</p> <p>5 Swtich (choice) { ----- ----- }</p> <p>Misspelling of keyword switch cannot be identified in LA phase because it is a valid identifier.</p> <p>Ex. For Exceeding length:</p> <p>In FORTRAN language identifier should have length of 10 characters. If exceeds it is an error.</p> <p>Ex. For Appearance of illegal characters :</p> <p>Consider <pre>printf("\n Hello India");\$</pre> - in this illegal character \$ appears at the end of the statement.</p> <p>2. Syntax Analysis phase Error: (2 Mark)</p> <p>If the token received LA phase violates the grammatical rules then syntactical errors get raised.</p> <p>Typical errors in this phase are</p> <ul style="list-style-type: none"> • Errors in structure Ex. C=a+b • Missing operator Ex. ab-c • Unbalanced parenthesis Ex. (a+(b-C) <p>3. Semantic Errors: (2 Mark)</p>

Meaning of the syntactical structure is detected during semantic analysis phase.

Typical errors in this phase are

- Incompatible types of operands
- Undeclared variables
- Not matching of actual arguments with formal arguments.

For example :

int a[10],b;

a=b;

It generates a semantic error.

PART * C

Explain in detail about programming language basics (15 Marks)

The compiler is also a software program that can be developed by any programming language like C or C++ etc. Let us see some of the basic concepts of programming language.

1. Static / Dynamic Decision Policy: (2 Marks)

The main concern that we should consider while designing for any programming language is what decision policy that a compiler should decide about the program. The decision policies taken by the compiler are of two types.

- Static Policy
- Dynamic Policy

If the programming language allows the compiler to decide about an issue at compile time, then it is said to be static policy. If the programming language allows the compiler to decide about an issue at run time, then the language uses Dynamic Policy.

2. Environments and states (2 Marks)

The association of names with locations in memory (the store) and then with values can be described by two mapping such as

The **Environment** is a mapping from names to locations in the store. The **state** is a mapping from locations in store to their values.

Example: Two declaration of the name i.

.....

```

int i;      /*global i */

.....
Void f(...) {
    int i; /* local i */

    ...
    i=3; /* use of local i */
    ...
}

.....
X=i+1; /* use of global i */

```

The environment and state mappings in above figure are dynamic, but there are few exceptions.

1. Static versus dynamic binding of names to locations.

Most binding of names to location is dynamic, some declaration such as the global i in the above example can be given a location in the store once and for all, the compiler generates object code.

2. Static versus dynamic binding of locations to values. (1 Mark)

The binding of location to values is generally dynamic as well, but declared exceptions are an exception.

3. Static scope : (1 Mark)

Most languages including C and its family use static scope. The scope rules for C are based on program structure. The scope of a declaration is determined implicitly by where the declaration appears in the program.

Later languages such as C++, Java and C# provides explicit control through the keywords like public, Private and protected.

4. Block Structure(2 Marks)

The block is the sequence of variable declarations and member function definitions. In C ,the block is represented by the braces '{' and '}'. In Algol, the block is represented by the keywords 'begin' and 'end'.

For example, the block in C++ is shown below,

Class ABC

{ Private: int a, b;

BLOCK B1

```

Public:      Void display()
{
A=10;          BLOCK B2
B=5;
Cout<<"a and b is:"<<a<<b;
}
}d;

```

5. Explicit Access Control: (2 Marks)

The access specifiers are used to indicate the scope of variables, in and out of the class. In object oriented languages like C++ or Java, the access specifiers are public, private and protected. These access specifiers provide encapsulation by restricting the access of the data members.

6. Dynamic Scope: (2 Marks)

If the scope of the variable is decided at runtime, then it is dynamic scope. The dynamic scope of the variable is decided by the parameter passing mechanisms. Calling the function with the parameters is said to be Procedure invocations. The parameters of the function can be of two types and explained with examples.

```

Actual Parameters
Formal parameters
Class ABC
{Public:
Int a,b,c;
Void display(int a,int b)//formal parameters
{ C=a+b; Cout<<c; }
}add;
Void main()
{ Add=new ABC();
Add.display(10,5);
}

```

7. Parameter Passing Mechanisms: (3 Marks)

The parameter passing mechanisms are of three types

- Call by value
- Call by reference
- Call by name

Call by value: A method is invoked by passing value as the actual parameters.

Call by reference: A method is invoked by passing the address of the variable as the actual parameter.

Call by name: A method is invoked by passing the address of the variable as the actual parameter.

JIT - 2106

UNIT II – LEXICAL ANALYSIS

Need and Role of Lexical Analyzer-Lexical Errors-Expressing Tokens by Regular Expressions-Converting Regular Expression to DFA- Minimization of DFA-Language for Specifying Lexical Analyzers-LEX-Design of Lexical Analyzer for a sample Language.

PART * A

Q.No.	Questions
1	<p>Differentiate lexeme, Pattern, Token.</p> <p>A Lexeme is a sequence of characters in the source program that is matched by the pattern for a token.</p> <p>1 A token is a pair consisting of a token name and an optional attribute value</p> <p>Pattern: A pattern is a description of the form that the lexemes of a token may take.</p>
2	<p>State the Error-recovery actions in a lexical analyser.</p> <ol style="list-style-type: none"> 1. Deleting an extraneous character 2. Inserting a missing character 3. Replacing an incorrect character by a correct character 4. Transposing two adjacent characters
3	<p>List the operations on languages.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Union - $L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$ <input type="checkbox"/> Concatenation – $LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$ <input type="checkbox"/> Kleene Closure – L^* (zero or more concatenations of L) <input type="checkbox"/> Positive Closure – $L^+ ($ one or more concatenations of L)
4	<p>Classify various error recovery strategies for a lexical analysis.</p> <p>Possible error recovery actions</p> <ul style="list-style-type: none"> <input type="checkbox"/> are: Panic mode recovery

	<input type="checkbox"/> Deleting an extraneous character <input type="checkbox"/> Inserting a missing character <input type="checkbox"/> Replacing an incorrect character by a correct character <input type="checkbox"/> Transposing two adjacent characters																					
5	<p>Why Lexical and syntax analyzers are separated. Justify.</p> <p>Reasons for separating the analysis phase into lexical and syntax analyzers:</p> <ul style="list-style-type: none"> • Simpler design. • Compiler efficiency is improved. <p>Compiler portability is enhanced</p>																					
6	<p>Define Lex Specifications</p> <p>A Lex program (.l file) consists of three parts:</p> <p>declarations</p> <pre>%%</pre> <p>translation rules</p> <pre>%%</pre> <p>auxiliary procedures</p>																					
7	<p>Differentiate compiler and interpreter.</p> <p>Compiler produces a target program. That is, it converts the source program into equivalent target program. Whereas an interpreter performs the operations implied by the source program. Interpreter produces the program output with line by line execution of the source program by taking the input.</p>																					
8	<p>Demonstrate some examples tokens, Sample lexeme, Pattern</p> <table border="1"> <thead> <tr> <th>TOKEN</th> <th>SAMPLE LEXEMES</th> <th>INFORMAL DESCRIPTION OF PATTERN</th> </tr> </thead> <tbody> <tr> <td>const</td> <td>const</td> <td>const</td> </tr> <tr> <td>if</td> <td>if</td> <td>if</td> </tr> <tr> <td>relation</td> <td><, <=, !=, >, >=</td> <td>< or <= or != or > or >= or ></td> </tr> <tr> <td>id</td> <td>pi, count, D2</td> <td>letter followed by letters and digits</td> </tr> <tr> <td>num</td> <td>3.1416, 0, 6.02E23</td> <td>any numeric constant</td> </tr> <tr> <td>literal</td> <td>"core dumped"</td> <td>any characters between " and " except "</td> </tr> </tbody> </table>	TOKEN	SAMPLE LEXEMES	INFORMAL DESCRIPTION OF PATTERN	const	const	const	if	if	if	relation	<, <=, !=, >, >=	< or <= or != or > or >= or >	id	pi, count, D2	letter followed by letters and digits	num	3.1416, 0, 6.02E23	any numeric constant	literal	"core dumped"	any characters between " and " except "
TOKEN	SAMPLE LEXEMES	INFORMAL DESCRIPTION OF PATTERN																				
const	const	const																				
if	if	if																				
relation	<, <=, !=, >, >=	< or <= or != or > or >= or >																				
id	pi, count, D2	letter followed by letters and digits																				
num	3.1416, 0, 6.02E23	any numeric constant																				
literal	"core dumped"	any characters between " and " except "																				

State the Issues in Lexical Analysis

1. **Simplicity of design** is the most important consideration. The separation of lexical and syntactic analysis often allows us to simplify at least one of these tasks.
2. **Compiler efficiency** is improved. A separate lexical analyzer allows us to apply specialized techniques that serve only the lexical task, not the job of parsing. In addition, specialized buffering techniques for reading input characters can speed up the compiler significantly.
3. Compiler portability is enhanced.

9

Construct a regular expression for an identifier.

An identifier is defined as a letter followed by zero or more letters or digits.

10

The regular expression for an identifier is given as

letter (letter | digit)*

PART * B

Describe roles and tasks of a lexical analyzer? (8 marks)

Main Task: Take a token sequence from the scanner and verify that it is a syntactically correct program. (2 Mark)

Secondary Tasks: Process declarations and set up symbol table information accordingly, in preparation for semantic analysis. Construct a syntax tree in preparation for intermediate code generation (2 Mark)

1

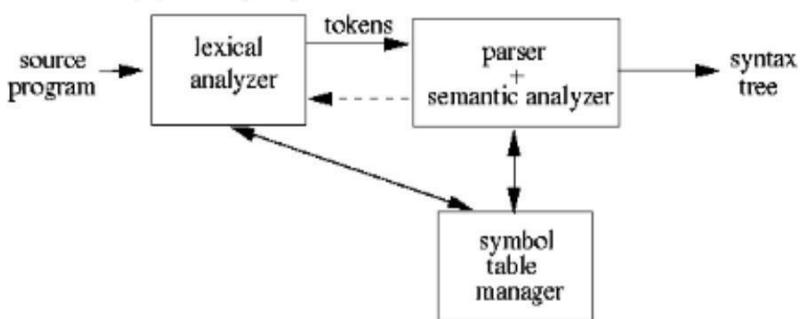


Diagram (4 Marks)

Describe an algorithm for Construction of an NFA from a Regular Expression (8 Marks)

Converting any regular expression to an NFA that defines the same language. The algorithm is syntax-directed, in the sense that it works recursively up the parse tree for the regular expression. For each sub expression the algorithm constructs an NFA with a single accepting state.

Algorithm: The McNaughton-Yamada-Thompson algorithm to convert a regular expression to an NFA.

2

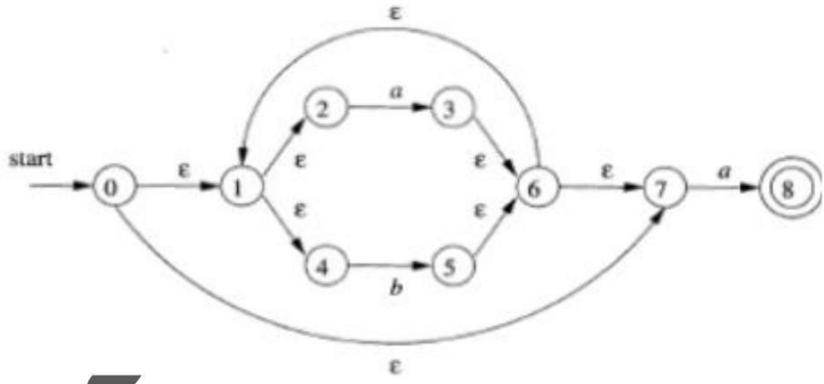
INPUT: A regular expression r over alphabet Σ .

OUTPUT: An NFA N accepting $L(r)$.

METHOD: Begin by parsing r into its constituent sub expressions. The rules for constructing an NFA consist of basis rules for handling sub expressions with no operators, and inductive rules for constructing larger NFA's from the NFA's for the immediate sub expressions of a given expression.

Construct the NFA using Thompson Construction for Regular Expression $(a/b)^* a$ (8 marks)

3



4

Construct DFA for an NFA using subset construction and describe the algorithm for subset construction. (13 Marks)

Algorithm (4 Marks)

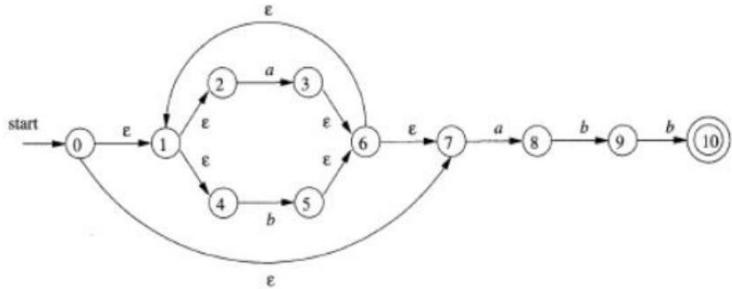
Problem (9 Marks)

```

push all states of  $T$  onto  $stack$ ;
initialize  $\epsilon$ -closure( $T$ ) to  $T$ ;
while (  $stack$  is not empty ) {
    pop  $t$ , the top element, off  $stack$ ;
    for ( each state  $u$  with an edge from  $t$  to  $u$  labeled  $\epsilon$  )
        if (  $u$  is not in  $\epsilon$ -closure( $T$ ) ) {
            add  $u$  to  $\epsilon$ -closure( $T$ );
            push  $u$  onto  $stack$ ;
        }
}

```

VB6

Figure 3.33: Computing ϵ -closure(T)Figure 3.34: NFA N for $(a|b)^*abb$

$$Dtran[A, a] = \epsilon\text{-closure}(move(A, a)) = \epsilon\text{-closure}(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$$

Let us call this set B , so $Dtran[A, a] = B$.

Now, we must compute $Dtran[A, b]$. Among the states in A , only 4 has a transition on b , and it goes to 5. Thus,

$$Dtran[A, b] = \epsilon\text{-closure}(\{5\}) = \{1, 2, 4, 6, 7\}$$

Let us call the above set C , so $Dtran[A, b] = C$.

NFA STATE	DFA STATE	a	b
$\{0, 1, 2, 4, 7\}$	A	B	C
$\{1, 2, 3, 4, 6, 7, 8\}$	B	B	D
$\{1, 2, 4, 5, 6, 7\}$	C	B	C
$\{1, 2, 4, 5, 6, 7, 9\}$	D	B	E
$\{1, 2, 3, 5, 6, 7, 10\}$	E	B	C

Figure 3.35: Transition table $Dtran$ for DFA D

Write short notes on Lex. Design of Lexical Analyzer for a sample Language (13 Marks)

Definition of LEX (2 Marks)

- 5 • There is a tool called Lex, or in a more recent implementation Flex, that allows one to specify a lexical analyzer by specifying regular expressions to describe patterns for tokens.
- The input notation for the Lex tool is referred to as the Lex language and the tool itself is

the Lex compiler.

- The Lex compiler transforms the input patterns into a transition diagram and generates code, in a file called `lex.y.y.c`, that simulates this transition diagram.
- An input file, which we call `lex.l`, is written in the Lex language and describes the lexical analyzer to be generated.
- The Lex compiler transforms `lex.l` to a C program, in a file that is always named `lex.y.y.c`.
- The latter file is compiled by the C compiler into a file called `a.out`, as always. The C-compiler output is a working lexical analyzer that can take a stream of input characters and produce a stream of tokens.

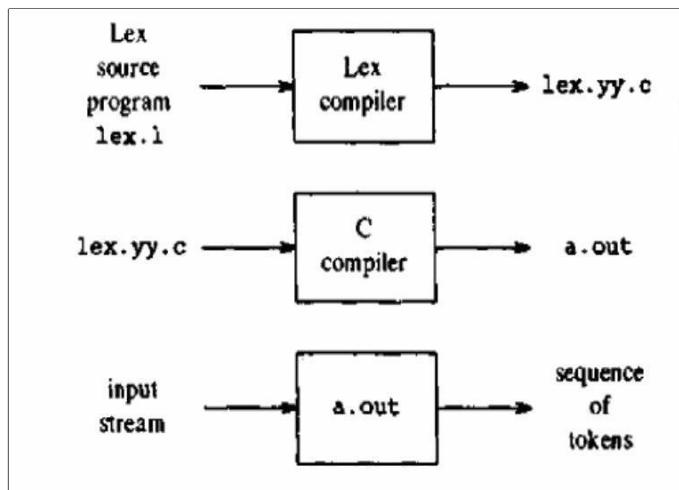


Diagram – (4 Marks)

Lex Specifications (2 Marks)

- A lex program has three parts
 - declarations**
 - %%**
 - translation rules**
 - %%**
 - auxiliary procedures**

The translation rules are statements of the form

$$\begin{array}{ll}
 p_1 & \{ \text{action}_1 \} \\
 p_2 & \{ \text{action}_2 \} \\
 \dots & \dots \\
 p_n & \{ \text{action}_n \}
 \end{array}$$

Design of Lexical Analyzer for a sample Language

In this %% - is called as separator

Example : 1 Lex program to find no of identifiers in the program (5 Marks)

```

// Declaration//

digit [0-9]

letter [A-Z a-z]

% { int count;

% }

%%

// transition rules//

/* match identifier */

{letter } ( {letter} | {digit})*   count++;

%%

// auxiliary procedure//

int main(void)

{yylex();

printf("number of identifiers : %d\n",count); return 0; }

```

PART * C

Construct DFA for the following regular expressions and minimize the DFA. (15 Marks)

1

(a|b)*.
(a*|b*)*

	Regular Expression to NFA (5 Marks) NFA TO DFA (6 Marks) DFA to Minimized DFA -(4 Marks)
--	--

JIT - 2106

UNIT III – SYNTAX ANALYSIS

Need and Role of the Parser-Context Free Grammars -Top Down Parsing -General Strategies- Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0)Item- Construction of SLR Parsing Table -Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer-YACC-Design of a syntax Analyzer for a Sample Language .

PART * A

Q.No.	Questions
1	<p>Define a context free grammar.</p> <p>A context free grammar G is a collection of the following</p> <ul style="list-style-type: none"> <input type="checkbox"/> V is a set of non-terminals <input type="checkbox"/> T is a set of terminals <input type="checkbox"/> S is a start symbol <input type="checkbox"/> P is a set of production rules G can be represented as <p>$G = (V, T, S, P)$</p> <p>Production rules are given in the following form</p> <p>Non terminal $\rightarrow (V \cup T)^*$</p>
2	<p>Define derivation.</p> <p>Derivation from S means generation of string w from S. For constructing derivation two things are important.</p> <ul style="list-style-type: none"> i) Choice of non terminal from several others. ii) Choice of rule from production rules for corresponding non terminal. Instead of choosing the arbitrary non terminal one can choose <ul style="list-style-type: none"> i) either leftmost derivation – leftmost non terminal in a sentinel form ii) or rightmost derivation – rightmost non terminal in a sentinel form

	Define ambiguous grammar. A grammar G is said to be ambiguous if it generates more than one parse tree for some sentence of language L(G). i.e. both leftmost and rightmost derivations are same for the given sentence
4	Summarize the properties of LR parser. <ol style="list-style-type: none"> 1. LR parsers can be constructed to recognize most of the programming languages for which the context free grammar can be written. 2. The class of grammar that can be parsed by LR parser is a superset of class of grammars that can be parsed using predictive parsers. 3. LR parsers work using non backtracking shift reduce technique yet it is efficient one
5	Classify LR parser. <ul style="list-style-type: none"> <input type="checkbox"/> SLR parser- simple LR parser <input type="checkbox"/> LALR parser- lookahead LR parser <input type="checkbox"/> Canonical LR parser
6	State problems with top down parsing? The following are the problems associated with top down parsing: <ul style="list-style-type: none"> <input type="checkbox"/> Backtracking <input type="checkbox"/> Left recursion <input type="checkbox"/> Left factoring <input type="checkbox"/> Ambiguity
7	Describe algorithm for FIRST and FOLLOW. FIRST <ol style="list-style-type: none"> 1. If X is terminal, then FIRST(X) IS {X}. 2. If $X \rightarrow \epsilon$ is a production, then add ϵ to FIRST(X). 3. If X is non terminal and $X \rightarrow Y_1, Y_2, \dots, Y_k$ is a production, then place a in FIRST(X) if for some i , a is in FIRST(Y_i) , and ϵ is in all of FIRST(Y₁),...,FIRST(Y_{i-1});

	FOLLOW
	<ol style="list-style-type: none"> 1. Place \$ in FOLLOW(S), where S is the start symbol and \$ is the input right endmarker. 2. If there is a production $A \rightarrow \alpha B \beta$, then everything in FIRST(β) except for ϵ is placed in FOLLOW(B). 3. If there is a production $A \rightarrow \alpha B$, or a production $A \rightarrow \alpha B \beta$ where FIRST(β) contains ϵ, then everything in FOLLOW(A) is in FOLLOW(B).
8	<p>Describe YACC. YACC is an automatic tool for generating the parser program. YACC stands for Yet Another Compiler Compiler which is basically the utility available from UNIX. Basically YACC is LALR parser generator. It can report conflict or ambiguities in the form of error messages</p>
9	<p>Describe handle pruning. A rightmost derivation in reverse can be obtained by handle pruning.</p> <p>If w is a sentence of the grammar at hand, then $w = \gamma_n$, where γ_n is the nth right-sentential form of some as yet unknown rightmost derivation</p> $S = \gamma_0 \Rightarrow \gamma_1 \dots \Rightarrow \gamma_{n-1} \Rightarrow \gamma_n = w$
10	<p>Define LR(0) items. An LR(0) item of a grammar G is a production of G with a dot at some position of the right side. Thus, production $A \rightarrow XYZ$ yields the four items</p> $\begin{aligned} & A \rightarrow .XYZ \\ & A \rightarrow X.YZ \\ & A \rightarrow XY.Z \\ & A \rightarrow XYZ. \end{aligned}$
	PART * B
1	<p>Construct a predictive parsing table for the grammar $E \rightarrow E + T / F$ $T \rightarrow T * F / F$ $F \rightarrow (E) / id$(13 Marks)</p> <p>Answer: The above grammar is left-recursive. (2 Marks) So eliminating left recursion we get,</p>

$$\begin{aligned}
 E &\rightarrow TE' \\
 \rightarrow & \\
 E' &\rightarrow +TE' \mid \epsilon \\
 \rightarrow & \\
 T &\rightarrow FT' \\
 \rightarrow & \\
 T' &\rightarrow *FT' \mid \epsilon \\
 \rightarrow & \\
 F &\rightarrow (E) \mid id
 \end{aligned}$$

Calculation of FIRST(X) and FOLLOW(A) - (5 Marks)

$$FIRST(+) = \{ + \}$$

$$FIRST(E) = FIRST(T) = FIRST(F) = \{ (, id \}$$

$$FIRST(*) = \{ * \}$$

$$FIRST(E') = \{ +, \epsilon \}$$

$$FIRST(()) = \{ (,) \}$$

$$FIRST(T') = \{ *, \epsilon \}$$

$$FIRST() = \{ \}$$

$$FIRST(id) = \{ id \}$$

$$FOLLOW(E) = FOLLOW(E') = \{) , \$ \}$$

$$FOLLOW(T) = FOLLOW(T') = \{ +,) , \$ \}$$

$$FOLLOW(F) = \{ +, *,), \$ \}$$

Parsing Table - (6 Marks)

Non-terminal	Input Symbol					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

CONSTRUCT LR(0) ITEMS for the grammar (13 Marks)

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$
Finding LR(0) items: (3 Marks)**Augmented Grammar – (2 Marks)****First and Follow –(2 Marks)****LR Parsing Table – (6 Marks)**

2

An $LR(0)$ item of a grammar G is a production of G with a dot at some position of the right side. For example, production $A \rightarrow XYZ$ yields the four items :

$$A \rightarrow . XYZ$$

$$A \rightarrow X . YZ$$

$$A \rightarrow XY . Z$$

$$A \rightarrow XYZ .$$

The given grammar is :

$G : E \rightarrow E + T$	----- (1)
$E \rightarrow T$	----- (2)
$T \rightarrow T * F$	----- (3)
$T \rightarrow F$	----- (4)
$F \rightarrow (E)$	----- (5)
$F \rightarrow id$	----- (6)

Step 1 : Convert given grammar into augmented grammar.

Augmented grammar :

- (r0) $E' \rightarrow E$
- (r1) $E \rightarrow E + T$
- (r2) $E \rightarrow T$
- (r3) $T \rightarrow T * F$
- (r4) $T \rightarrow F$
- (r5) $F \rightarrow (E)$
- (r6) $F \rightarrow id$

$$\text{FOLLOW}(E) = \{ \$, , + \}$$

$$\text{FOLLOW}(T) = \{ \$, +, , * \}$$

$$\text{FOLLOW}(F) = \{ \$, +, , * \}$$

JIT

V106

Closure	ACTION					GOTO	
	Terminals			Non Terminals		E	T
Id	+	*	()	S		
I ₀	s ₅			s ₄		1	2
I ₁		s ₆			ACC		
I ₂		r ₂	s ₇		r ₂	r ₂	
I ₃		r ₄	r ₄		r ₄	r ₄	
I ₄	s ₅			s ₄		8	2
I ₅		r ₆	r ₆		r ₆	r ₆	
I ₆	s ₅			s ₄			9
I ₇	s ₅			s ₄			
I ₈	s ₆				s ₁₁		
I ₉		r ₁	s ₇		r ₁	r ₁	
I ₁₀		r ₃	r ₃		r ₃	r ₃	
I ₁₁		r ₅	r ₅		r ₅	r ₅	

Design of a syntax Analyzer for a Sample Language.

Yacc provides a general tool for imposing structure on the input to a computer program.

The Yacc user prepares a specification of the input process; this includes rules describing the input structure, code to be invoked when these rules are recognized, and a low-level routine to do the basic input.

3

Yacc then generates a function to control the input process. This function, called a parser, calls the user-supplied low-level input routine (the lexical analyzer) to pick up the basic items (called tokens) from the input stream.

These tokens are organized according to the input structure rules, called grammar rules; when one

of these rules has been recognized, then user code supplied for this rule, an action, is invoked; actions have the ability to return values and make use of the values of other actions.

The yacc command converts a context-free grammar into a set of tables for a simple automaton that executes an LALR(1) parsing algorithm. The grammar may be ambiguous. Specified precedence rules are used to break ambiguities. The output file, y.tab.c, must be compiled by the C compiler to produce a function yyparse(). This program must be loaded with the lexical analyzer program, yylex(), as well as main() and yyerror(), an error handling routine. These routines must be supplied by the user. The Lex (1) command is useful for creating lexical analyzers usable by yacc.

Like lex, yacc has its own specification language. A yacc specification is structured along the same lines as a Lex specification.

```
%{
/* C declarations and includes */

%}

/* Yacc token and type declarations */
%%

/* Yacc Specification
   in the form of grammar rules like this:
*/
symbol  :  symbols tokens
          { $$ = my_c_code($1); }
;
%%
```

Example : To Write a program to implement a calculator using Yacc Tool.

ALGORITHM:

STEP 1: Define and include necessary ‘C’ declarations and token definitions

STEP 2: Define the translation rules and the input structure specifications for the grammar rules

STEP 3: Define the functions that are invoked in the rules

STEP 4: Read the input

STEP 5: If the given input matches with the defined rules , the yacc tool executes the respective actions.

STEP 6: End

PROGRAM:

//Program to implement calculator using yacc Tool

```
%{
#define YYSTYPE double
%}
```

```
%token NUMBER
%left '+'
%left '*'
%
list: /*nothing*/
    |list '\n'
    |list expr '\n' {printf("\t%.8g\n",$2);}
    ;
expr: NUMBER {$$=$1;}
    |expr '+' expr {$$=$1+$3;}
    |expr '-' expr {$$=$1-$3;}
    |expr '*' expr {$$=$1*$3;}
    |expr '/' expr {$$=$1/$3;}
    |'(' expr ')' {$$=$2;}
    ;
%
#include<stdio.h>
#include<ctype.h>
char *progname;
int lineno=1;
main(argc,argv)
char *argv[];
{
    progname=argv[0];
    yyparse();
}

yylex()
{
    int c;
    while((c=getchar())==' '||c=='\t')
        ;
    if(c==EOF)
        return 0;
    if(c=='.'||isdigit(c))
    {
        ungetc(c,stdin);
        scanf("%lf",&yylval);
        return NUMBER;
    }
    if(c=='\n')
        lineno++;
    return(c);
}
```

How to execute:

\$yacc ss.y (create a file called y.tab.c)

\$cc y.tab.c -ly

\$.a.out

SAMPLE INPUT & OUTPUT

Input:

5 + 6

Output:

11

Explain Context Free Grammar with Example . (10 Marks)

4 CFG Definition – 2 Marks

Derivation and Types – RMD,LMD – 4 marks

Parse Tree – 2 Marks

Ambiguous Grammar – 2 Marks

Construct Stack Implementation of shift reduce parsing for the following grammar E->E+E

E->E*E

E->(E)

E->id and the input string is id1+id2*id3

2106
(8 Marks)

Stack Implementation – 6 Marks

Actions – 2 Marks

PART * :

Check whether the given grammar is LL(1) or not. (15 Marks)

$\begin{array}{l} S \rightarrow iEtS \mid iEtSeS \mid a \\ E \rightarrow b \end{array}$

1

Elimination of Left Factoring -2 Marks

First and Follow – 6 Marks

Predictive Parsing Table – 5 Marks

The above grammar has a left factor.
Hence left-factoring the grammar we get ,
 $S \xrightarrow{i} EtSS' | a$
 $S' \xrightarrow{} eS | \epsilon$
 $E \xrightarrow{} b$

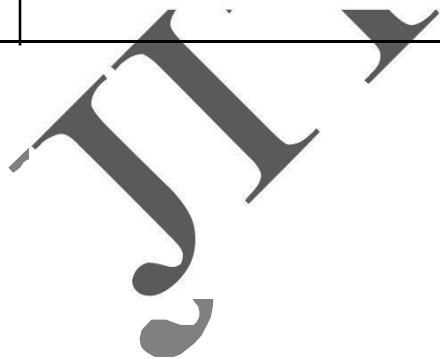
Calculation of FIRST(X) and FOLLOW(A)

$$\begin{array}{ll} \text{FIRST}(i)=\{i\} & \text{FIRST}(S)=\{ i, a \} \\ \text{FIRST}(t)=\{t\} & \text{FIRST}(S')=\{ e, \epsilon \} \\ \text{FIRST}(e)=\{e\} & \text{FIRST}(E)=\{ b \} \\ \text{FIRST}(a)=\{a\} & \\ \text{FIRST}(b)=\{b\} & \end{array}$$

$$\begin{array}{l} \text{FOLLOW}(S)=\{ e, \$ \} \\ \text{FOLLOW}(S')=\{ e, \$ \} \\ \text{FOLLOW}(E)=\{ t \} \end{array}$$

Parsing Table

Non-terminal	Input Symbol					
	a	b	e	i	t	s
S	$S \xrightarrow{} a$			$S \xrightarrow{i} EtSS'$		
S'			$S' \xrightarrow{} eS$			$S' \xrightarrow{} \epsilon$
E	$E \xrightarrow{} b$					



106

Unit IV: SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT

Syntax directed Definitions-Construction of Syntax Tree-Bottom-up Evaluation of S-Attribute Definitions- Design of predictive translator -Type Systems-Specification of a simple type checker Equivalence of Type Expressions-Type Conversions.

RUN-TIME ENVIRONMENT: Source Language Issues-Storage Organization-Storage Allocation Parameter Passing-Symbol Tables-Dynamic Storage Allocation-Storage Allocation in FORTAN

PART *A

Q.NO	QUESTIONS
1.	<p>Define syntax directed definition. BTL1</p> <ul style="list-style-type: none"> ✓ Syntax directed definition is a generalization of context free grammar in which each grammar production $X \rightarrow \alpha$ is associated with it a set of semantic rules of the form $a := f(b_1, b_2, \dots, b_k)$, where a is an attribute obtained from the function f.
2.	<p>List the benefits of intermediate code generation BTL1</p> <ul style="list-style-type: none"> ✓ A Compiler for different machines can be created by attaching different back end to the existing front ends of each machine. ✓ A Compiler for different source languages can be created by proving different front ends for corresponding source languages to existing back end. ✓ A machine independent code optimizer can be applied to intermediate code in order to optimize the code generation
3.	<p>Explain the two notations for attaching the semantic rule. BTL1</p> <p>There are two notations for attaching semantic rules:</p> <ul style="list-style-type: none"> ✓ Syntax Directed Definitions. High-level specification hiding many implementation details (also called Attribute Grammars). ✓ Translation Schemes. More implementation oriented: Indicate the order in which semantic rules are to be evaluated.
4.	<p>Discuss the two types of attributes? BTL1</p> <p>We distinguish between two kinds of attributes:</p> <ul style="list-style-type: none"> ✓ Synthesized Attributes. They are computed from the values of the attributes of the children nodes. ✓ Inherited Attributes. They are computed from the values of the attributes of both the siblings and the parent nodes.
5.	<p>What is L-attributed definition? BTL1</p> <p>Definition: A SDD is <i>L-attributed</i> if each inherited attribute of X_i in the RHS of $A \to X_1 : : : X_n$ depends only on</p> <ul style="list-style-type: none"> ✓ attributes of $X_1; X_2; : : : ; X_i$ (symbols to the left of X_i in the RHS) ✓ inherited attributes of A
6.	<p>List the 3 ways of Storage Organization. BTL1</p> <p><i>There are 3 ways</i></p>

	<ul style="list-style-type: none"> ✓ Fixed-size objects can be placed in predefined locations ✓ Run-time stack and heap ✓ Activation records
7.	<p>List the Content of Activation record. BTL1</p> <hr/> <p>Returned value</p> <hr/> <p>Actual parameters</p> <hr/> <p>Optional control link</p> <hr/> <p>Optional access link</p> <hr/> <p>Saved machine status</p> <hr/> <p>Local data</p> <hr/> <p>Temporaries</p>
8.	<p>How the value of inherited attribute is computed? BTL2</p> <p>It was computed from the value of attributes at the siblings and parent of that node.</p>
9.	<p>Define Symbol Table. BTL1</p> <p>A symbol table is a major data structure used in a compiler. Associates attributes with identifiers used in a program. For instance, a type attribute is usually associated with each identifier. A symbol table is a necessary component Definition (declaration) of identifiers appears once in a program.</p>
10.	<p>What is dynamic storage-allocation strategy. BTL1</p> <p>The data area requirements for a program are not known entirely at compilation time. The size and number of each object need not be known at compile time; however, they must be known at run time when a block is entered. Similarly more than one occurrence of a data object is allowed, provided that each new occurrence is initiated at run time when a block is entered.</p>
11.	<p>List the various ways to pass a parameter in a function? BTL1</p> <hr/> <ul style="list-style-type: none"> ✓ Call by value ✓ Call by reference ✓ Copy-restore ✓ Call by name
12.	<p>List the functions used to create the nodes of syntax trees? BTL1</p> <hr/> <ul style="list-style-type: none"> ✓ Mknode (op, left, right) ✓ Mkleaf (id, entry) ✓ Mkleaf (num, val)
13	<p>Type checker verifies that the type of a construct (constant, variable, array, list, object) matches what is expected in its usage context.</p>
14	<p>What are static and dynamic errors? BTL1</p> <hr/> <ul style="list-style-type: none"> ✓ Static error: It can be detected at compile time. Eg: Undeclared identifiers. ✓ Dynamic errors: It can be detected at run time. Eg: Type checking
	<p>Define activation trees. BTL1</p>

15	<p>A recursive procedure p need not call itself directly; p may call another procedure q, which may then call p through some sequence of procedure calls. We can use a tree called an activation tree, to depict the way control enters and leaves activation. In an activation tree</p> <ul style="list-style-type: none"> ✓ Each node represents an activation of a procedure, ✓ The root represents the activation of the main program ✓ The node for a is the parent of the node for b if and only if control flows from activation a to b, and ✓ The node for a is to the left of the node for b if and only if the lifetime of a occurs before the lifetime of b.
16.	<p>What are the advantages of compile time checking? BTL1</p> <ul style="list-style-type: none"> ✓ It can catch many common errors. ✓ Static checking is desired when speed is important, since it can result in faster code that does not perform any type checking during execution.
17.	<p>What are the advantages of the dynamic checking? BTL1</p> <ul style="list-style-type: none"> ✓ It usually permits the programmer to be less concerned with types. Thus, it frees the programmer. ✓ It may be required in some cases like array bounds check, which can be performed only during execution. ✓ It can give clearer code. ✓ It may result in more robust code by ensuring thorough checking of values for the program identifiers during execution.
18.	<p>Define type systems. BTL1</p> <p>Type system of a language is a collection of rules depicting the type expression assignments to program objects. An implementation of a type system is called a type checker.</p>
19.	<p>Write Static vs. Dynamic Type Checking BTL1</p> <ul style="list-style-type: none"> ✓ Static: Done at compile time (e.g., Java) ✓ Dynamic: Done at run time (e.g., Scheme) ✓ Sound type system is one where any program that passes the static type checker cannot contain run-time type errors. Such languages are said to be strongly typed.
20.	<p>The size of a data object and constraints on its position in memory must be known at compile time.</p> <ul style="list-style-type: none"> ✓ Recursive procedure is restricted. ✓ Data structures cannot be created dynamically.
21	<p>Stack allocation is based on the idea of a control stack; storage is organized as a stack, and activation records are pushed and popped as activations begin and end respectively.</p>
22	<p>List the fields in activation record .(Nov/Dec 2014) BTL1</p> <ul style="list-style-type: none"> ✓ Actual parameters ✓ Returned Values ✓ Control link ✓ Access link ✓ Saved machine status ✓ Local data

	Temporaries
23	What is dangling references? BTL1 Whenever storage can be de-allocated, the problem of dangling references arises. A dangling reference occurs when there is a reference to storage that has been de allocated.
24	Write a 3-address code for; $x = *y$; $a = \&x$. (April/May 2015) BTL1 t1:=*y x:=t1 t1:=&x a:=t1
25	How the value of synthesized attribute is computed? BTL1 It was computed from the values of attributes at the children of that node in the parse tree.
1	<p>Construct parse tree, syntax tree and annotated parse tree for the input string is $5*6+7$. (13M) BTL6</p> <p>Answer Page:- 368 Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>Parse tree (5M)</p> <pre> graph TD E --> T1 E --> T2 T1 --> D1 T1 --> D2 D1 --> N1 D2 --> N2 T2 --> D3 style E fill:none,stroke:none style T1 fill:none,stroke:none style T2 fill:none,stroke:none style D1 fill:none,stroke:none style D2 fill:none,stroke:none style D3 fill:none,stroke:none N1[3] --- D1 N2[5] --- D2 N3[4] --- D3 </pre> <p>Syntax tree (3M)</p> <pre> graph TD P[+] --- N1[3] P --- N2[*] P --- N3[4] N2 --- N4[5] N2 --- N5[6] style P fill:none,stroke:none style N1 fill:none,stroke:none style N2 fill:none,stroke:none style N3 fill:none,stroke:none style N4 fill:none,stroke:none style N5 fill:none,stroke:none </pre> <p>Annotated parse tree (5M)</p>

	<pre> graph TD L --> E17[E.val=17] L --> Return[return] E17 --> E5[E.val=5] E17 --> Star[*] E5 --> T5[T.val=5] Star --> T12[T.val=12] Star --> Plus[+] T12 --> T3[T.val=3] T12 --> F4[F.val=4] T3 --> F3[F.val=3] F3 --> DL5[digit.lexval=5] F4 --> DL4[digit.lexval=4] </pre>
2	<p>Explain 1)Synthesized attribute 2)inherited attribute with suitable examples. (13M) BTL2</p> <p>Answer Page:358 - Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>Synthesized attributes (7M)</p> <ul style="list-style-type: none"> ✓ A synthesized attribute for a nonterminal A at a parse-tree node N is defined by a semantic rule associated with the production at N. ✓ Note that the production must have A as its head. ✓ A synthesized attribute at node N is defined only in terms of attribute values at the children of N and at N itself. ✓ These attributes get values from the attribute values of their child nodes. <p>Example</p> <ol style="list-style-type: none"> 1. $S \rightarrow ABC$ If S is taking values from its child nodes (A, B, C), then it is said to be a synthesized attribute, as the values of ABC are synthesized to S. 2. $E \rightarrow E + T$ The parent node E gets its value from its child node E and T. <pre> E → E1 + T { E.val = E1.val + T.val; } E → T { E.val = T.val; } T → (E) { T.val = E.val; } T → digit { T.val = digit.lexval; } </pre> <p>Inherited attributes (6M)</p> <ul style="list-style-type: none"> ✓ An inherited attribute for a nonterminal B at a parse-tree node N is defined by a semantic rule associated with the production at the parent of N. ✓ Note that the production must have B as a symbol in its body. ✓ An inherited attribute at node N is defined only in terms of attribute values at JV's parent, N itself, and N's siblings. ✓ Inherited attributes can take values from parent and/or siblings. <p>Example</p> $S \rightarrow ABC$ <p>A can get values from S, B and C. B can take values from S, A, and C. Likewise, C can take values from S, A, and B.</p>

Example

	$\begin{array}{ll} E \rightarrow T A & \{ E.\text{node} = A.s; \\ & A.i = T.\text{node}; \} \\ A \rightarrow + T A_1 & \{ A_1.i = \text{Node}(+, A.i, T.\text{node}); \\ & A.s = A_1.s; \} \\ A \rightarrow e & \{ A.s = A.i; \} \\ \\ T \rightarrow (E) & \{ T.\text{node} = E.\text{node}; \} \\ \\ T \rightarrow \text{id} & \{ T.\text{node} = \text{Leaf}(\text{id}, \text{id}.entry); \} \end{array}$								
3	<p>(i) Explain about Activation records in detail. (7M) BTL2 Answer Page:433- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>Activation record: (3M)</p> <ul style="list-style-type: none"> ✓ Procedure calls and returns are usually managed by a run time stack called the <i>control stack</i>. ✓ Each live activation has an activation record on the control stack, with the root of the activation tree at the bottom, the latter activation has its record at the top of the stack. The contents of the activation record vary with the language being implemented. T <p>Diagram with explanation: (4M)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Activation record</th> </tr> </thead> <tbody> <tr> <td>Actual Parameters</td> </tr> <tr> <td>Returned Values</td> </tr> <tr> <td>Control Link</td> </tr> <tr> <td>Access Links</td> </tr> <tr> <td>Saved Machine states</td> </tr> <tr> <td>Local Data</td> </tr> <tr> <td>Temporaries</td> </tr> </tbody> </table> <p>(ii) Describe about type checking and type systems.(6M) BTL2</p> <p>Answer Page:540- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>TYPE CHECKING (4M)</p> <p>A compiler must check that the source program follows both syntactic and semantic conventions of the source language.</p> <p>This checking, called <i>static checking</i>, detects and reports programming errors.</p> <p>Some examples of static checks:</p> <ul style="list-style-type: none"> ✓ Type checks – A compiler should report an error if an operator is applied to an incompatible operand. Example: If an array variable and function variable are added together. ✓ Flow-of-control checks – Statements that cause flow of control to leave a construct must have some place to which to transfer the flow of control. Example: An error 	Activation record	Actual Parameters	Returned Values	Control Link	Access Links	Saved Machine states	Local Data	Temporaries
Activation record									
Actual Parameters									
Returned Values									
Control Link									
Access Links									
Saved Machine states									
Local Data									
Temporaries									

	<p>occurs when an enclosing statement, such as break, does not exist in switch statement.</p> <p>A type checker verifies that the type of a construct matches that expected by its context. For example : arithmetic operator <i>mod</i> in Pascal requires integer operands, so a type checker verifies that the operands of <i>mod</i> have type integer.</p> <pre> graph LR TS[token stream] --> P[parser] P -- syntax --> TC[typechecker] TC -- syntax --> ICG[intermediate code generator] ICG -- intermediate --> IC[intermediate] </pre>
	<p>TYPE SYSTEMS (2M)</p> <ul style="list-style-type: none"> ✓ The design of a type checker for a language is based on information about the syntactic constructs in the language, the notion of types, and the rules for assigning types to language constructs. ✓ For example : “ if both operands of the arithmetic operators of +,- and * are of type integer, then the result is of type integer ”
4	<p>What are different storage allocation strategies? Explain. (May/June,2016) (April/May,2017) (13M) BTL1</p> <p>Answer Page:524- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>STORAGE ALLOCATION STRATEGIES (3M)</p> <p>The different storage allocation strategies are :</p> <ul style="list-style-type: none"> ✓ Static allocation – lays out storage for all data objects at compile time ✓ Stack allocation – manages the run-time storage as a stack. ✓ Heap allocation – allocates and deallocates storage as needed at run time from a data area known as heap. <p>STATIC ALLOCATION (4M)</p> <ul style="list-style-type: none"> ✓ In static allocation, names are bound to storage as the program is compiled, so there is no need for a run-time support package. ✓ Since the bindings do not change at run-time, everytime a procedure is activated, its names are bound to the same storage locations. Therefore values of local names are <i>retained</i> across activations of a procedure. That is, ✓ when control returns to a procedure the values of the locals are the same as they were when control left the last time. ✓ From the type of a name, the compiler decides the amount of storage for the name and decides where the activation records go. At compile time, we can fill in the addresses at which the target code can find the data it operates on. <p>STACK ALLOCATION OF SPACE (3M)</p> <ul style="list-style-type: none"> ✓ All compilers for languages that use procedures, functions or methods as units of

	<p>user defined actions manage at least part of their run-time memory as a stack.</p> <ul style="list-style-type: none"> ✓ Each time a procedure is called , space for its local variables is pushed onto a stack, and when the procedure terminates, that space is popped off the stack. ✓ Calling sequences: ✓ Division of tasks between caller and callee ✓ Variable length data on stack: <p>HEAP ALLOCATION (3M)</p> <p>Stack allocation strategy cannot be used if either of the following is possible :</p> <ul style="list-style-type: none"> ✓ The values of local names must be retained when an activation ends. ✓ A called activation outlives the caller. ✓ Heap allocation parcels out pieces of contiguous storage, as needed for activation records or other objects. ✓ Pieces may be deallocated in any order, so over the time the heap will consist of alternate areas that are free and inuse.
5	<p>Explain about Source language issues in detail with examples. BTL2 (13M)</p> <p>Answer Page: 575- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>The source language issues are (3M)</p> <ul style="list-style-type: none"> ✓ Procedures ✓ Activation Trees ✓ Control stack ✓ Scope of Declaration ✓ Binding of names <p>PROCEDURES: (2M)</p> <ul style="list-style-type: none"> ✓ A <i>procedure definition</i> is a declaration that associates an identifier with a statement. ✓ The identifier is the <i>procedure name</i>, and the statement is the <i>procedure body</i>. <p>Example:</p> <pre>procedure readarray; var i : integer; begin for i := 1 to 9 do read(a[i]) end;</pre> <p>When a procedure name appears within an executable statement, the procedure is said to be <i>called</i> at that point.</p> <p>ACTIVATION TREES: (2M)</p> <p>An <i>activation tree</i> is used to depict the way control enters and leaves activations. In an activation tree,</p> <ol style="list-style-type: none"> 1. Each node represents an activation of a procedure. 2. The root represents the activation of the main program. 3. The node for <i>a</i> is the parent of the node for <i>b</i> if and only if control flows from activation <i>a</i> to <i>b</i>. 4. The node for <i>a</i> is to the left of the node for <i>b</i> if and only if the lifetime of <i>a</i> occurs before

the lifetime of b .

CONTROL STACK:

(2M)



A *control stack* is used to keep track of live procedure activations. The idea is to push the node for an activation onto the control stack as the activation begins and to pop the node when the activation ends.



The contents of the control stack are related to paths to the root of the activation tree. When node n is at the top of control stack, the stack contains the nodes along the path from n to the root.

THE SCOPE OF A DECLARATION:

(2M)



A declaration is a syntactic construct that associates information with a name. Declarations may be explicit, such as:

Example:

var i : integer ;

or they may be implicit. Example, any variable name starting with I is assumed to denote an integer.

The portion of the program to which a declaration applies is called the *scope* of that declaration.

BINDING OF NAMES:

(2M)



The term *environment* refers to a function that maps a name to a storage location.



The term *state* refers to a function that maps a storage location to the value held there.

environment state name storage value

When an *environment* associates storage location s with a name x , we say that x is *bound* to s . This association is referred to as a *binding* of x .

PART * C

- 1 Specify a type checker which can handle expressions, statements and functions. (Nov/Dec, 2017) (13M) BTL5

Answer Page:450- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.

A SIMPLE LANGUAGE

(2M)

Consider the following grammar:

$P \rightarrow D ; E$

$D \rightarrow D ; D \mid id : T$

$T \rightarrow char \mid integer \mid array [num] of T \mid \uparrow T$

$E \rightarrow literal \mid num \mid id \mid E \ mod \ E \mid E [E] \mid E \uparrow$

TRANSLATION SCHEME:

(2M)

$P \rightarrow D , E$

$D \rightarrow D ; D$

{ addtype (id.entry , T.type)}

$D \rightarrow id : T$

$T \rightarrow char \{ T.type := char \}$

$T \rightarrow integer \{ T.type := integer \}$

$T \rightarrow \uparrow T1 \{ T.type := pointer(T1.type) \}$

$T \rightarrow array [num] of T1 \{ T.type := array (1... num.val , T1.type) \}$

In the above language,

- There are two basic types : char and integer ;
- *type_error* is used to signal errors;
- the prefix operator \uparrow builds a pointer type. Example , \uparrow integer leads to the type expression **pointer (integer)**.

TYPE CHECKING OF EXPRESSIONS (3M)

In the following rules, the attribute *type* for E gives the type expression assigned to the expression generated by E.

1. $E \rightarrow \text{literal} \{ E.\text{type} := \text{char} \}$

$E \rightarrow \text{num} \{ E.\text{type} := \text{integer} \}$

Here, constants represented by the tokens **literal** and **num** have type *char* and *integer*.

2. $E \rightarrow \text{id} \{ E.\text{type} := \text{lookup} (\text{id}.entry) \}$

lookup (e) is used to fetch the type saved in the symbol table entry pointed to by e.

3. $E \rightarrow E_1 \text{ mod } E_2 \{ E.\text{type} := \text{if } E_1.\text{type} = \text{integer}$

and $E_2.\text{type} = \text{integer}$ **then**

integer **else** *type_error* }

The expression formed by applying the mod operator to two subexpressions of type *integer* has type *integer*; otherwise, its type is *type_error*.

4. $E \rightarrow E_1 [E_2] \{ E.\text{type} := \text{if } E_2.\text{type} = \text{integer} \text{ and}$

$E_1.\text{type} = \text{array}(s,t) \text{ then } t$

else *type_error* }

In an array reference $E_1 [E_2]$, the index expression E_2 must have type *integer*. The result is the element type *t* obtained from the type *array(s,t)* of E_1 .

5. $E \rightarrow E_1 \uparrow \{ E.\text{type} := \text{if } E_1.\text{type} = \text{pointer}(t) \text{ then}$

t **else** *type_error* }

The postfix operator \uparrow yields the object pointed to by its operand. The type of $E \uparrow$ is the type *t* of the object pointed to by the pointer E .

TYPE CHECKING OF STATEMENTS (3M)

Statements do not have values; hence the basic type *void* can be assigned to them. If an error is detected within a statement, then *type_error* is assigned.

Translation scheme for checking the type of statements:

1. Assignment statement:

$S \rightarrow \text{id} := E \{ S.\text{type} := \text{if } \text{id}.\text{type} = E.\text{type} \text{ then void}$

else *type_error* }

2. Conditional statement:

$S \rightarrow \text{if } E \text{ then } S_1 \{ S.\text{type} := \text{if } E.\text{type} = \text{boolean} \text{ then}$

$S_1.\text{type} \text{ else } \text{type_error}$ }

3. While statement:

$S \rightarrow \text{while } E \text{ do } S_1 \{ S.\text{type} := \text{if } E.\text{type} = \text{boolean} \text{ then } S_1.\text{type}$

else *type_error* }

4. Sequence of statements:

$S \rightarrow S_1 ; S_2 \{ S.\text{type} := \text{if } S_1.\text{type} = \text{void} \text{ and } S_2.\text{type} = \text{void} \text{ then void}$

else *type_error* }

else type_error }

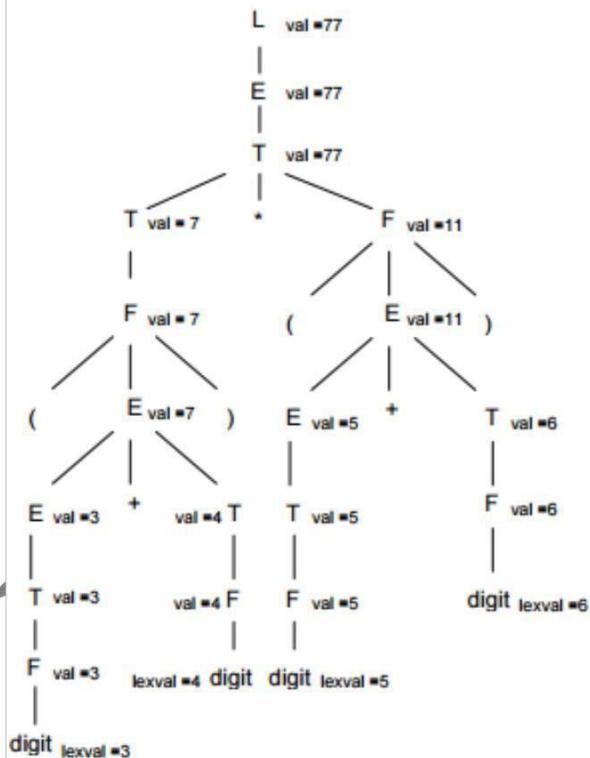
TYPE CHECKING OF FUNCTIONS (3M)

The rule for checking the type of a function application is :

$E \rightarrow E_1(E_2) \{ E.type := \text{if } E_2.type = s \text{ and } E_1.type = s \rightarrow t \text{ then } t \text{ else type_error } \}$

- 2 (i) Given the Syntax-Directed Definition below with the synthesized attribute val, draw the annotated parse tree for the expression $(3+4)^*(5+6)$. (7M) BTL6

$L \rightarrow E \quad L.val = E.val$
 $E \rightarrow T \quad E.val = T.val$
 $E \rightarrow E_1 + T \quad E.val = E_1.val + T.val$
 $T \rightarrow F \quad T.val = F.val$
 $T \rightarrow T_1 * F \quad T.val = T_1.val *$
 $F.val \rightarrow (E) \quad F.val = E.val$
 $F \rightarrow \text{digit} \quad F.val = \text{digit.lexval}$



- (ii) Explain the various structures that are used for the symbol table constructions. (April/may 2014) (6M) BTL2

Answer Page:-463 Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.



A separate array 'arr_lexemes' holds the character string forming an identifier. The

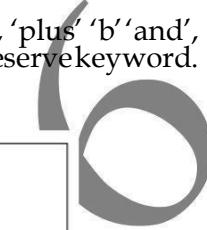
string is terminated by an end-of-string character, denoted by EOS, that may not appear in identifiers.

- ✓ Each entry in symbol-table array 'arr_symbol_table' is a record consisting of two fields, as "lexeme pointer", pointing to the beginning of a lexeme, and token.
- ✓ Additional fields can hold attribute values. 0th entry is left empty, because lookup return 0 to indicate that there is no entry for a string.
- ✓ The 1st, 2nd, 3rd, 4th, 5th, 6th, and 7th entries are for the 'a', 'plus' 'b' 'and', 'c', 'minus', and 'd' where 2nd, 4th and 6th entries are for reserve keyword.

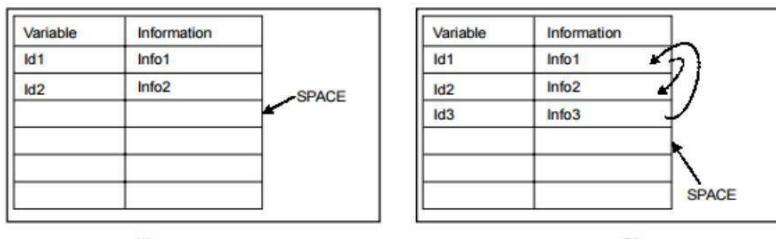
List

(2M)

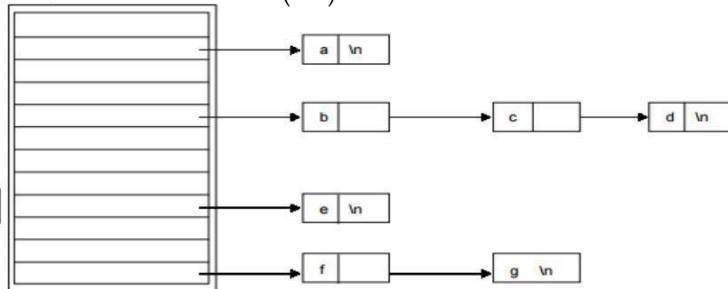
Variable	Information(type)	Space (byte)
a	Integer	2
b	Float	4
c	Character	1
d	Long	4

**Self Organizing List**

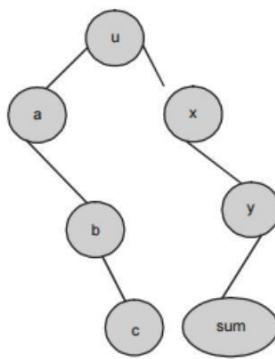
(2M)

**Hash Table**

(1M)

**Search Tree**

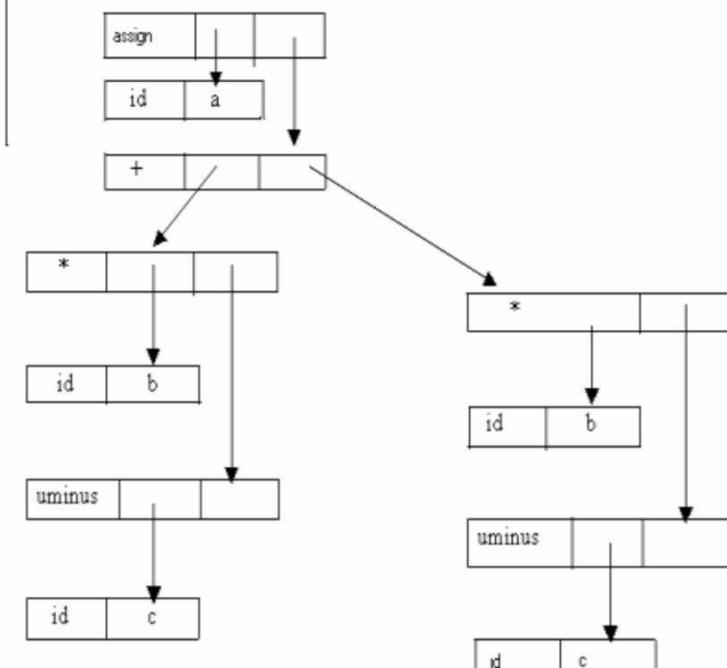
(1M)



- 3 Construct a syntax directed definition and syntax tree for assignment statements. $S \rightarrow id$
- $$\begin{aligned}
 &:= E \\
 &E \rightarrow E+EE \\
 &\quad \rightarrow E*EE \\
 &\quad (E) \\
 &E \rightarrow id
 \end{aligned}$$

(May/June,2016) (13M) BTL 6

0	id	b	
1	id	c	
2	uminus	1	
3	*	0	2
4	id	b	
5	id	c	
6	uminus	5	
7	*	4	6
8	+	3	7
9	id	a	
10	assign	9	8
11		



Syntax tree for assignment statements are produced by the syntax directed definition	
Production	Semantic Rule
$S \rightarrow id := E$	$S.nptr := mknode('assign', mkleaf(id, id.place), E.nptr)$
$E \rightarrow E1 + E2$	$E.nptr := mknode('+', E1.nptr, E2.nptr)$
$E \rightarrow E1 * E2$	$E.nptr := mknode('*', E1.nptr, E2.nptr)$
$E \rightarrow - E1$	$E.nptr := mkunode('uminus', E1.nptr)$
$E \rightarrow (E1)$	$E.nptr := E1.nptr$
$E \rightarrow id$	$E.nptr := mkleaf(id, id.place)$

JIT-2106

Unit V: CODE OPTIMIZATION AND ZCODE GENERATION	
Principal Sources of Optimization-DAG- Optimization of Basic Blocks-Global Data Flow Analysis Efficient Data Flow Algorithms-Issues in Design of a Code Generator - A Simple Code Generator Algorithm.	
Q.NO	QUESTIONS
1.	<p>List the properties that a code generator should possess. BTL1</p> <ul style="list-style-type: none"> ✓ The code generator should produce the correct and high quality code. In other words, the code generated should be such that it should make effective use of the resources of the target machine. ✓ Code generator should run efficiently.
2.	<p>List the terminologies used in basic blocks. BTL1</p> <ul style="list-style-type: none"> ✓ Define and use – the three address statement $a := b + c$ is said to define a and to use b and c. ✓ Live and dead – the name in the basic block is said to be live at a given point if its value is used after that point in the program. And the name in the basic block is said to be dead at a given point if its value is never used after that point in the program.
3.	<p>Define flow graph. BTL1 A flow graph is a directed graph in which the flow control information is added to the basic blocks.</p> <ul style="list-style-type: none"> ✓ The nodes to the flow graph are represented by basic blocks ✓ The block whose leader is the first statement is called initial block. ✓ There is a directed edge from block B_1 to block B_2 if B_2 immediately follows B_1 in the given sequence. We can say that B_1 is a predecessor of B_2.
4.	<p>Describe DAG. Mention its applications. BTL2 Directed acyclic graph(DAG) is a useful data structure for implementing transformations on basic blocks.</p> <p>DAG is used in</p> <ul style="list-style-type: none"> ✓ Determining the common sub-expressions. ✓ Determining which names are used inside the block and computed outside the block. ✓ Determining which statements of the block could have their computed value outside the block. ✓ Simplifying the list of quadruples by eliminating the common sub-expressions and not performing the assignment of the form $x := y$ unless and until it is a must
5.	<p>Define peephole optimization. BTL1 Peephole optimization is a simple and effective technique for locally improving target code. This technique is applied to improve the performance of the target program by examining the short sequence of target instructions and replacing these instructions by shorter or faster sequence</p>
6.	<p>List the characteristics of peephole optimization. BTL1</p> <ul style="list-style-type: none"> ✓ Redundant instruction elimination

	<ul style="list-style-type: none"> ✓ Flow of control optimization ✓ Algebraic simplification ✓ Use of machine idioms
7.	<p>Describe How do you calculate the cost of an instruction? BTL2</p> <p>The cost of an instruction can be computed as one plus cost associated with the source and destination addressing modes given by added cost.</p> <p>MOV R0,R1 1 MOV R1,M 2 SUB 5(R0),*10(R1) 3</p>
8.	<p>Define basic block. BTL1</p> <p>A basic block is a sequence of consecutive statements in which flow of control enters at the beginning and leaves at the end without halt or possibility of branching.</p> <p>Eg.</p> <pre>t1:=a*5 t2:=t1+7 t3:=t2-5 t4:=t1+t3 t5:=t2+b</pre>
9.	<p>List the issues to be considered while applying the techniques for code optimization. BTL1</p> <ul style="list-style-type: none"> ✓ The semantic equivalence of the source program must not be changed. ✓ The improvement over the program efficiency must be achieved without changing the algorithm of the program
10.	<p>Describe the basic goals of code movement? BTL2</p> <ul style="list-style-type: none"> ✓ To reduce the size of the code i.e. to obtain the space complexity. ✓ To reduce the frequency of execution of code i.e. to obtain the time complexity
11.	<p>Discuss about machine dependent and machine independent optimization. BTL3</p> <ul style="list-style-type: none"> ✓ The machine dependent optimization is based on the characteristics of the target machine for the instruction set used and addressing modes used for the instructions to produce the efficient target code. ✓ The machine independent optimization is based on the characteristics of the programming languages for appropriate programming structure and usage of efficient arithmetic properties in order to reduce the execution time
12.	<p>List the different data flow properties. BTL1</p> <ul style="list-style-type: none"> ✓ Available expressions ✓ Reaching definitions ✓ Live variables ✓ Busy variables
13	<p>In dynamic scoping a use of non-local variable refers to the non-local data declared in most recently called and still active procedure. Therefore each time new findings are set up for local names called procedure. In dynamic scoping symbol tables can be required at run time</p>
14	<p>Describe code motion. BTL2</p> <p>Code motion is an optimization technique in which amount of code in a loop is decreased. This transformation is applicable to the expression that yields the same result independent of the number of times the loop is executed. Such an expression is placed before the loop</p>
	<p>List the properties of optimizing compiler? BTL1</p>

	The source code should be such that it should produce minimum amount of target code. There should not be any unreachable code.
15	<p>Dead code should be completely removed from source language.</p> <p>The optimizing compilers should apply following code improving transformations on source language.</p> <ul style="list-style-type: none"> ✓ common subexpression elimination ✓ dead code elimination ✓ code movement ✓ strength reduction
16.	<p>What are the principle sources of optimization? BTL1</p> <p>The principle sources of optimization are, Optimization consists of detecting patterns in the program and replacing these patterns by equivalent but more efficient constructs. The richest source of optimization is the efficient utilization of the registers and instruction set of a machine.</p>
17.	<p>What is the step takes place in peephole optimization? BTL1</p> <p>It improves the performance of the target program by examining a short sequence of target instructions. It is called peephole. Replace this instructions by a shorter or faster sequence whenever possible. It is very useful for intermediate representation..</p>
18.	<p>What are the characteristics of peephole optimization? BTL1</p> <ul style="list-style-type: none"> ✓ Redundant instruction elimination. ✓ Flow of control optimization ✓ Algebraic simplifications ✓ Use of machine idioms
19.	<p>List the criteria for selecting a code optimization technique. BTL1</p> <p>The criteria for selecting a good code optimization technique are, It should capture most of the potential improvement without an unreasonable amount of effort. It should preserve the meaning of the program. It should reduce the time or space taken by the object program..</p>
20.	<p>List any two structure preserving transformations adopted by the optimizer? BTL1</p> <p>The structure preserving transformations adopted by the optimizer are, Basic blocks.-Flow graphs.</p>
21	<p>What are dominators? BTL1</p> <p>A node of flow graph is said to be a dominator, i.e one node dominates the other node if every path from the initial node of the flow graph to that node goes through the first node.(d Dom n).when d-node dominates n-node.</p>
22	<p>What do you mean by data flow equations? BTL1</p> <p>A typical equation has the form $\text{out}[s] = \text{gen}[s] \cup (\text{in}[s] - \text{kill}[s])$</p> <p>It can be read as information at the end of a statement is either generated within the statement or enters at the beginning and is not killed as control flows through the statement.</p> <p>State the meaning of $\text{in}[s]$, $\text{out}[s]$, $\text{kill}[s]$, $\text{gen}[s]$ in $[s]$-The set of definitions reaching the beginning of S.$\text{out}[s]$-End of S.$\text{gen}[s]$-The set of definitions generated by S.$\text{kill}[s]$-The set of definitions that never reach the end of S.</p>
23	<p>Define loop unrolling with example.(Nov/Dec 2017) BTL1</p> <p>Loop overhead can be reduced by reducing the number of iterations and replicating the body of the loop.</p> <p>Example:</p> <p>In the code fragment below, the body of the loop can be replicated once and the number of iterations can be reduced from 100 to 50.</p> <pre>for (i = 0; i < 100; i++) g();</pre> <p>Below is the code fragment after loop unrolling. for (i = 0; i < 100; i += 2)</p> <pre>{ g();</pre>

	g (); }
24	<p>What are the rules to find “ leader” in basic block? BTL1</p> <ul style="list-style-type: none"> ✓ It is the first statement in a basic block is a leader. ✓ Any statement which is the target of a conditional or unconditional goto is a leader. ✓ Any statement which immediately follows a conditional goto is a leader.
25	<p>It automatically detects common sub expression. We can determine which identifiers have their values used in the block.</p> <ul style="list-style-type: none"> ✓ We can determine which statements compute values, and which could be used outside the block . It reconstruct a simplified list of quadruples taking advantage of common sub expressions and not performs assignments of the form $a=b$ unless necessary
	PART* B
1	<p>Explain the issues in design of a code generator.(13M) (BTL 2)</p> <p>Answer Page:510- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>While the details are dependent on the target language and the operating system, issues such as</p> <ul style="list-style-type: none"> Input to Code Generator Target Programs Memory Management Instruction Selection Register Allocation Choice of Evaluation Order Approaches to Code Generation. <p>(i) Input to Code Generator (2M)</p> <ul style="list-style-type: none"> several choices for intermediate languages: ✓ Linear representations such as postfix notation ✓ Three-address representations such as quadruples ✓ Virtual machine representations such as stack machine code ✓ Graphical representations such as trees and DAGs <p>(ii) Target Programs (2M)</p> <ul style="list-style-type: none"> The output of the code generator is the target program. The output of code generator may be in different form: <ul style="list-style-type: none"> - Absolute machine language - Relocatable machine language - Assembly language. ✓ can be placed in a fixed location in memory and immediately executed. ✓ Small program can be compiled and executed quickly. <p>Relocatable Machine Language:</p> <ul style="list-style-type: none"> ✓ Producing it as output allows subprograms to be compiled separately. ✓ A set of relocatable object modules can be linked together and loaded for execution by a linking loader. <p>Assembly Language:</p> <ul style="list-style-type: none"> ✓ It makes the process of code generation easier. ✓ It can generate symbolic instructions and use the macro facilities of the assembler to help generate code. <p>(iii)Memory Management (2M)</p>

- ✓ Mapping of names in the source program to addresses of data objects in run-time memory is done by the front-end and code generator.
- ✓ If the machine code is being generated, labels in the three-address statements have to be converted to addresses of instructions.
- ✓ The instruction set of the target machine determines the difficulty of instruction selection.
- ✓ The uniformity, completeness of the instruction set, instruction speed and machine idioms are important factors.

$x := y + z$, where x , y and z are statically allocated, can be translated into code sequence

```
MOV y, R0      /* load y into register R0 */
ADD z, R0      /* add z to R0 */
MOV R0, x      /* store R0 into x */
```

But , this kind of statement-by-statement code generation often produces poor code. For example, the sequence of statements

$a := b + c$

$d := a + e$

would be translated into

```
MOV b, R0
ADD c, R0
```

MOV R0, a

```
MOV a, R0
ADD e, R0
MOV R0, d
```

The fourth statement is redundant and so is the third if ‘a’ is not subsequently used.

- The quality of the generated code is determined by its speed and size.
- ✓ Instructions involving register operands are usually shorter and faster than those involving operands in memory.
- ✓ Efficient utilization of registers is particularly important in generating good code.

(vi) Choice of Evaluation Order (2M)

The order in which computations are performed can affect the efficiency of the target code.

(vii) Approaches to Code Generation (1M)

The important criterion for a code generator is that it produce correct code.

- ✓ Correctness takes on special significance because of the number of special cases that a code generator might face.

2 Describe the Optimization / Transformations on Basic Blocks. (13M) BTL2

Answer Page:521- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.

There are two important classes of local transformations/ optimizations that can be applied to basic blocks.

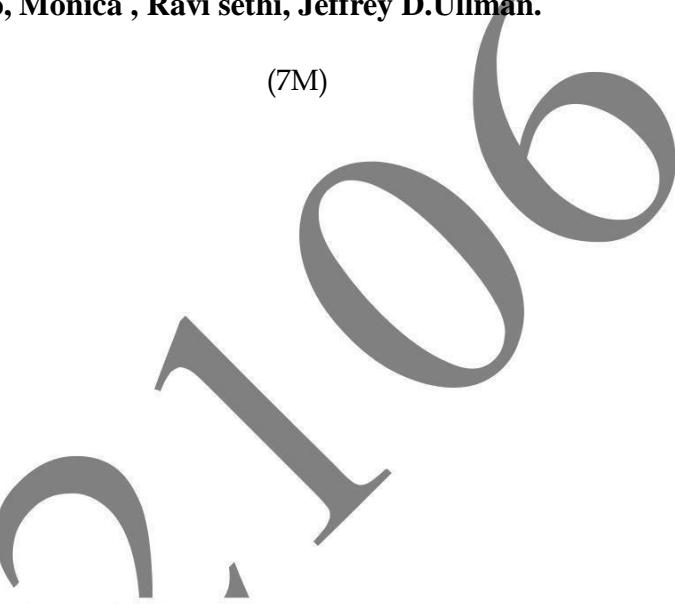
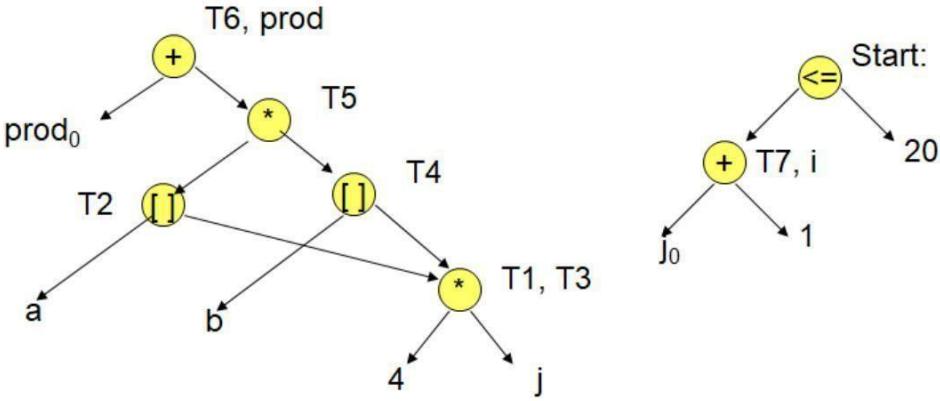
- ✓ Structure-Preserving Transformations
- ✓ Algebraic Transformations

Structure-Preserving Transformations (7M)

- ✓ Common Sub-Expression Elimination
- ✓ Dead-Code Elimination

	<ul style="list-style-type: none"> ✓ Renaming of Temporary Variables ✓ Interchange of Two Independent Adjacent Statements. <p>Algebraic Transformations (8M)</p> <p>Algebraic transformations can be used to change the set of expressions computed by a basic block into an algebraically equivalent set.</p> <p>Eg: Statements such as</p> <p>$x := x + 0$ (or) $x := x * 1$</p> <p>can be eliminated from a basic block without changing the set of expressions it computes.</p> <p>The exponentiation operator in the statement</p> <p>$x := y^{**} 2$</p> <p>usually requires a function call to implement.</p> <p>Using an algebraic transformation, this statement can be replaced by the cheaper, but equivalent statement</p> <p>$x := y * y$.</p>
3	<p>Explain the simple code generator with a suitable example (May/June,2016) (13M) BTL2</p> <p>Answer Page:542- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>Definition: (1M)</p> <ul style="list-style-type: none"> ✓ A code generator generates target code for a sequence of three- address statements and effectively uses registers to store operands of the statements. <p>For example: consider the three-address statement $a := b+c$</p> <p>It can have the following sequence of codes:</p> <p>ADD R_j, R_i Cost = 1 // if R_i contains b and R_j contains c (or)</p> <p>ADD c, R_i Cost = 2 // if c is in a memory location (or)</p> <p>MOV c, R_j Cost = 3 // move c from memory to R_j and add</p> <p>ADD R_j, R_i</p> <p>Register and Address Descriptors: (3M)</p> <ul style="list-style-type: none"> ✓ A register descriptor is used to keep track of what is currently in each registers. ✓ An address descriptor stores the location where the current value of the name can be found at run time. <p>A code-generation algorithm: (3M)</p> <ul style="list-style-type: none"> ✓ The algorithm takes as input a sequence of three -address statements constituting a basic block. For each three-address statement of the form $x := y \text{ op } z$, perform the following actions: <ul style="list-style-type: none"> Invoke a function <i>getreg</i> to determine the location L where the result of the computation $y \text{ op } z$ should be stored. Consult the address descriptor for y to determine y'', the current location of y. Prefer the register for y'' if the value of y is currently both in memory and a register. If the value of y is not already in L, generate the instruction MOV y', L to place a copy of y in L. <p>Generating Code for Assignment Statements: (3M)</p> <p>The assignment $d := (a-b) + (a-c) + (a-c)$ might be translated into the following three address code sequence:</p> <p>$t := a - b$</p>

	<p>✓ $u := a - c$ ✓ $v := t + u$ ✓ $d := v + u$ with d live at the end.</p> <p>Code sequence for the example is:</p> <table border="1"> <thead> <tr> <th>Statements</th><th>Code Generated</th><th>Register descriptor</th><th>Address descriptor</th></tr> </thead> <tbody> <tr> <td>$t := a - b$</td><td>MOV a, R₀</td><td>R₀ contains t t in R₀</td><td></td></tr> <tr> <td>SUB b, R₀</td><td></td><td></td><td></td></tr> <tr> <td>$u := a - c$</td><td>MOV a , R₁</td><td>R₀ contains t t in R₀</td><td></td></tr> <tr> <td>SUB c , R₁</td><td></td><td>R₁ contains u u in R₁</td><td></td></tr> <tr> <td>$v := t + u$</td><td>ADD R₁, R₀</td><td>R₀ contains v u in R₁</td><td></td></tr> <tr> <td>R_1 contains u v in R₀</td><td></td><td></td><td></td></tr> <tr> <td>$d := v + u$</td><td>ADD R₁, R₀</td><td>R₀ contains d d in R₀</td><td></td></tr> <tr> <td>MOV R₀, d</td><td></td><td></td><td></td></tr> <tr> <td>d in R₀ and memory</td><td></td><td></td><td></td></tr> </tbody> </table>	Statements	Code Generated	Register descriptor	Address descriptor	$t := a - b$	MOV a, R ₀	R ₀ contains t t in R ₀		SUB b, R ₀				$u := a - c$	MOV a , R ₁	R ₀ contains t t in R ₀		SUB c , R ₁		R ₁ contains u u in R ₁		$v := t + u$	ADD R ₁ , R ₀	R ₀ contains v u in R ₁		R_1 contains u v in R ₀				$d := v + u$	ADD R ₁ , R ₀	R ₀ contains d d in R ₀		MOV R ₀ , d				d in R ₀ and memory			
Statements	Code Generated	Register descriptor	Address descriptor																																						
$t := a - b$	MOV a, R ₀	R ₀ contains t t in R ₀																																							
SUB b, R ₀																																									
$u := a - c$	MOV a , R ₁	R ₀ contains t t in R ₀																																							
SUB c , R ₁		R ₁ contains u u in R ₁																																							
$v := t + u$	ADD R ₁ , R ₀	R ₀ contains v u in R ₁																																							
R_1 contains u v in R ₀																																									
$d := v + u$	ADD R ₁ , R ₀	R ₀ contains d d in R ₀																																							
MOV R ₀ , d																																									
d in R ₀ and memory																																									
4	<p>Discuss about the following: (13M)</p> <p>i) Copy Propagation (3M) ii) Dead-code Elimination (5M) iii) Code motion. (5M)</p> <p>BTL2 Answer Page:26- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p>Copy Propagation (3M)</p> <p>Statement of form $f := g$ is called a copy statement</p> <ul style="list-style-type: none"> ✓ Idea is to use g instead of f in subsequent statements ✓ Doesn't help by itself, but can combine with other transformations to help eliminate code: <pre> B5 x := t3 a[t2] := t5 a[t4] := x goto B2 </pre> <p>copy propagation</p> <pre> B2 x := t3 a[t2] := t5 a[t4] := t3 goto B2 </pre> <p>Dead-Code Elimination (5M)</p> <p>Variable that is no longer live (subsequently used) is called dead.</p> <p>Copy propagation often turns copy statement into dead code:</p> <pre> B5 x := t3 a[t2] := t5 a[t4] := x goto B2 </pre> <p>copy propagation</p> <pre> B2 x := t3 a[t2] := t5 a[t4] := t3 goto B2' </pre> <p>dead-code elimination</p> <pre> B2' a[t2] := t5 a[t4] := t3 goto B2' </pre> <p>Code Motion (5M)</p> <ul style="list-style-type: none"> • Expression whose value doesn't change inside loop is called a loop-invariant • Code motion moves loop-invariants outside loop 																																								

	<pre>while (i <= limit-2) ↓ code motion t = limit - 2; while (i <= t)</pre>
5	<p>Construct the DAG for the following Basic Block. (April/May,2017)(13M)BTL 6 Answer Page:533- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.</p> <p style="text-align: right;">(7M)</p> <p>✓ t1:=4*i ✓ t2:=a [t1] ✓ t3:=4*i ✓ t4:=b[t3] ✓ t5:=t2*t4 ✓ t6:=prod +t5 prod :=t6 ✓ t7:=i+1 ✓ i:=t7 ✓ if i<=20 goto (1)</p> <p>DAG structure with steps : (8M)</p>  <ul style="list-style-type: none"> Common subexpression identified 

PART * C	
1	<p>Construct Basic Block, Flow graph and DAG for the following code segment. (13M) BTL 6 Answer Page:525- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman. begin</p> <pre>prod := 0; i := 1; do begin</pre>

```

prod := prod + a[i] * b[i];
i := i + 1;
end
while i <= 20
end

```

Basic Blocks: (1M)

- ✓ A basic block is a sequence of consecutive statements in which flow of control enters at the beginning and leaves at the end without halt or possibility of branching except at the end.
- ✓ A name in a basic block is said to be **live** at a given point if its value is used after that point in the program, perhaps in another basic block.

Algorithm for partitioning a sequence of three-address statements into basic blocks: (3M)

- ✓ **Input:** A sequence of three-address statements.
- ✓ **Output:** A list of basic blocks with each three-address statement in exactly one block.

Method:

1. Determine the set of leaders, the first statements of basic blocks.

Rules to follow:

- (i) The first statement is a leader.
- (ii) Any statement that is the target of a conditional or unconditional goto is a leader.
- (iii) Any statement that immediately follows a goto or conditional goto statement is a leader.

2. For each leader, its basic block consists of the leader and all statements up to but not including the next leader or the end of the program.

```

prod := 0
i := 1

```

```

t1 := 4 * i
t2 := a [t1]
t3 := 4 * i
t4 := b [t3]
t5 := t2 * t4
t6 := prod + t5
prod := t6
t7 := i + 1
i := t7
if i <= 20 goto (3)

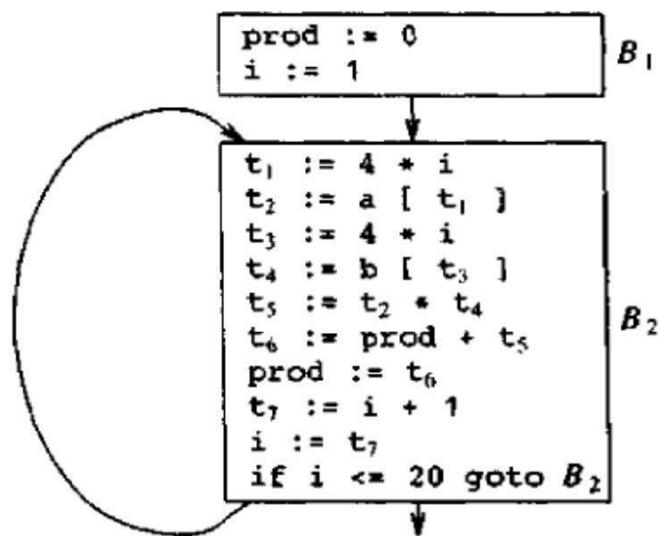
```

(3M)

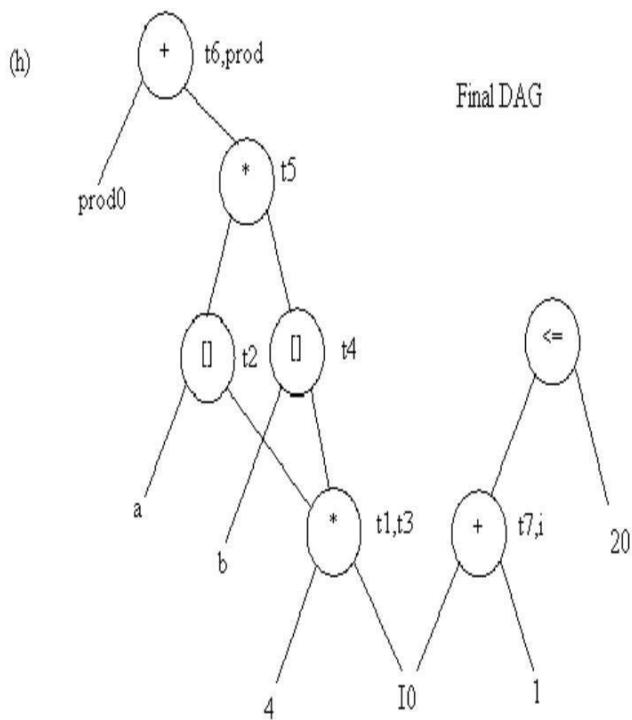
Flow Graphs

(3M)

- ✓ Add the flow-of-control information to the set of basic blocks making up a program by
- ✓ constructing a directed graph called a flow graph.
The nodes of the flow graph are the basic blocks.

**DAG REPRESENTATION OF BASIC BLOCKS**

(3M)



- 2 Explain the principle sources of optimization in detail. (April/May,2017) (Nov/Dec, 2017)
(13M) BTL 2

Answer Page:560- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.

Transformations can be (1M)
Local : look within basic block
Global : look across blocks

Transformations should preserve function of program.

- ✓ Function-preserving transformations include

Common sub expression elimination

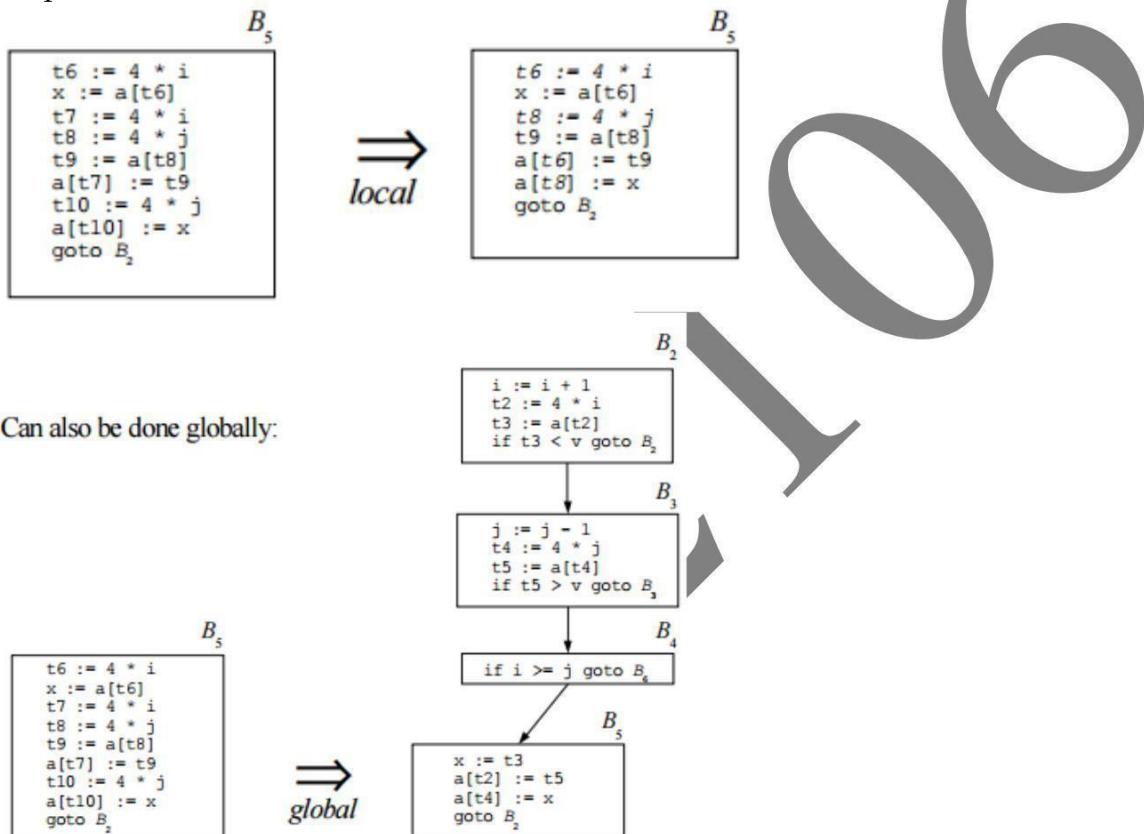
Copy propagation

Dead-code elimination

Constant-folding

Common Sub expression Elimination (3M) Occurrence of expression E is called common sub expression if

- E was previously computed, and values of variables in E have not changed since previous Computation



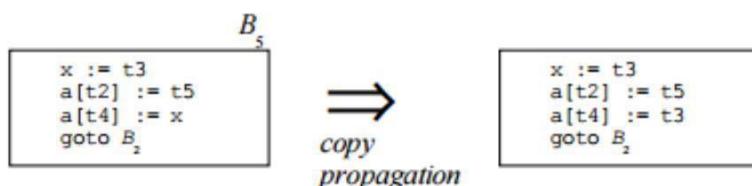
Copy Propagation

(2M)

- ✓ Statement of form $f := g$ is called a copystatement

- ✓ Idea is to use g instead of f in subsequent statements

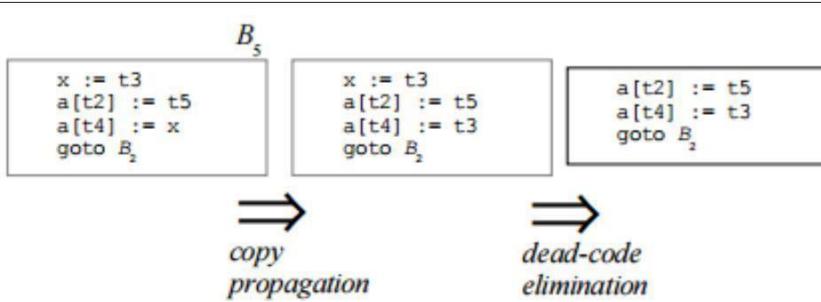
- ✓ Doesn't help by itself, but can combine with other transformations to help eliminate code:



Dead-Code Elimination

(2M)

- ✓ Variable that is no longer live (subsequently used) is called dead.
- ✓ Copy propagation often turns copy statement into dead code:

**Loop Optimizations** (2M)

Biggest speedups often come from moving code out of inner loop

Three techniques

- ✓ Code motion
- ✓ Induction-variable elimination
- ✓ Reduction in strength

Code Motion (2M)

- ✓ Expression whose value doesn't change inside loop is called a loop-invariant
- ✓ Code motion moves loop-invariants outside loop

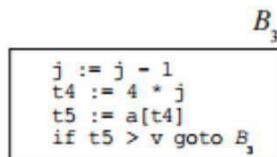
```
while (i <= limit-2)
```

\Downarrow code motion

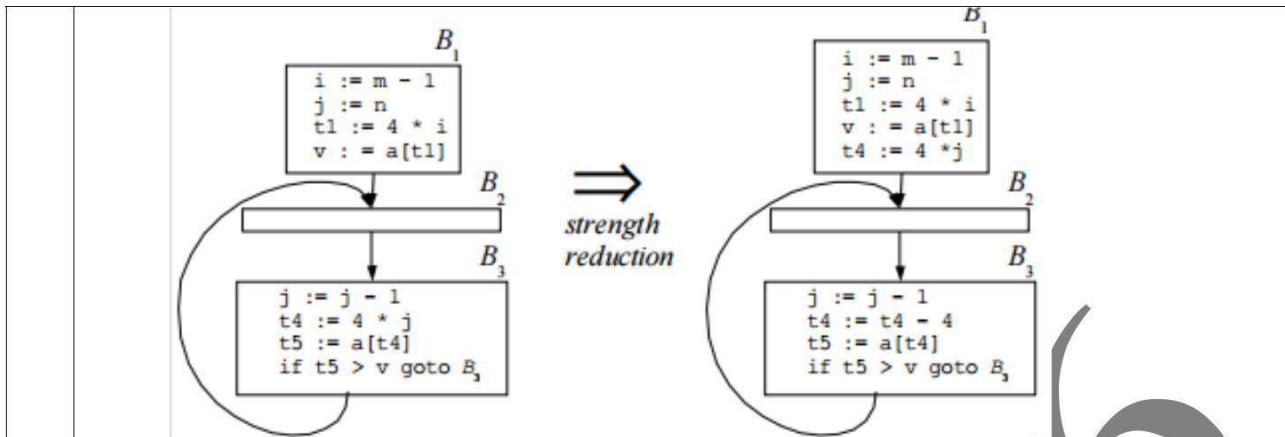
```
t = limit - 2;
while (i <= t)
```

Induction Variables and Reduction in Strength (2M)

- ✓ Variables that remain "in lock step" with each other inside a loop are called induction variables
- ✓ E.g., decreasing array byte-offset index by 4 as loop variable decreases by 1:



- ✓ Addition is like multiplication, "reduced in strength" (less costly)
- ✓ Exploit induction variables and reduction -in-strength to make loop code more efficient



- 3 Explain global data flow analysis with necessary equations (Nov/Dec 2016)(13M) BTL2
Consider the following language:

Answer Page: 597- Alfred V. Aho, Monica , Ravi sethi, Jeffrey D.Ullman.

Introduction: (3M)

$S ::= \text{id} := \text{expression} \mid S;S \mid \text{if expression then } S \text{ else } S \mid \text{do } S \text{ while expression}$
We define the following equations to compute reaching definitions. In these equations we compute reaching definitions for statements instead of blocks.

Equations: (10M)

We use the following terms:

- $\text{gen}[S]$ is the set of definitions “generated by S ”.
- $\text{kill}[S]$ is the set of definitions “killed” by S .
- $\text{in}[S]$ is the set of definitions reaching S (or the top of S)
- $\text{out}[S]$ is the set of definitions that reach the bottom of S .

1. When S has the form $a := \text{expression}$ and label d : $\text{gen}[S] = \{d\}$

$\text{kill}[S] = \text{Da} - \{d\}$. Da is the set of all definitions of a .

$\text{out}[S] = \text{gen}[S] + (\text{in}[S] - \text{kill}[S])$

2. When S is of the form $S_1; S_2$:

$\text{gen}[S] = \text{gen}[S_2] + (\text{gen}[S_1] - \text{kill}[S_2])$

$\text{kill}[S] = \text{kill}[S_2] + (\text{kill}[S_1] - \text{gen}[S_2])$

$\text{in}[S_1] = \text{in}[S]$

$\text{in}[S_2] = \text{out}[S_1]$

$\text{out}[S] = \text{out}[S_2]$

3. When S is of the form $\text{if } ... \text{ then } S_1 \text{ else } S_2$

$\text{gen}[S] = \text{gen}[S_1] + \text{gen}[S_2]$

$\text{kill}[S] = \text{kill}[S_1] \cap \text{kill}[S_2]$

$\text{in}[S_1] = \text{in}[S]$

$\text{in}[S_2] = \text{in}[S]$

$\text{out}[S] = \text{out}[S_1] + \text{out}[S_2]$

4. When S is of the form $\text{do } S_1 \text{ while } ...$:

$\text{gen}[S] = \text{gen}[S_1]$

$\text{kill}[S] = \text{kill}[S_1]$

$\text{in}[S_1] = \text{in}[S] + \text{gen}[S_1]$

$\text{out}[S] = \text{out}[S_1]$

CS8603 . DISTRIBUTED SYSTEMS**L T P C 3 0 0 3****OBJECTIVES:**

- To understand the foundations of distributed systems.
- To learn issues related to clock Synchronization and the need for global state in distributed systems.
- To learn distributed mutual exclusion and deadlock detection algorithms.
- To understand the significance of agreement, fault tolerance and recovery protocols in Distributed Systems.
- To learn the characteristics of peer-to-peer and distributed shared memory systems.

UNIT I INTRODUCTION 9

Introduction: Definition –Relation to computer system components –Motivation –Relation to parallel systems –Message-passing systems versus shared memory systems –Primitives for distributed communication –Synchronous versus asynchronous executions –Design issues and challenges. A model of distributed computations: A distributed program –A model of distributed executions –Models of communication networks –Global state – Cuts –Past and future cones of an event –Models of process communications. Logical Time: A framework for a system of logical clocks –Scalar time –Vector time – Physical clock synchronization: NTP.

UNIT II MESSAGE ORDERING & SNAPSHOTS 9

Message ordering and group communication: Message ordering paradigms –Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system –Group communication –Causal order (CO) - Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions –Snapshot algorithms for FIFO channels

UNIT III DISTRIBUTED MUTEX & DEADLOCK 9

Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport’s algorithm – Ricart-Agrawala algorithm – Maekawa’s algorithm – Suzuki-Kasami’s broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries – Models of deadlocks – Knapp’s classification – Algorithms for the single resource model, the AND model and the OR model.

UNIT IV RECOVERY & CONSENSUS 9

Checkpointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated checkpointing algorithm – Algorithm for asynchronous checkpointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure – free system – Agreement in synchronous systems with failures.

UNIT V P2P & DISTRIBUTED SHARED MEMORY 9

Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry. Distributed shared memory: Abstraction and advantages – Memory consistency models –Shared memory Mutual Exclusion.

TOTAL: 45 PERIODS**OUTCOMES:**

At the end of this course, the students will be able to:

- Elucidate the foundations and issues of distributed systems
- Understand the various synchronization issues and global state for distributed systems.
- Understand the Mutual Exclusion and Deadlock detection algorithms in distributed systems
- Describe the agreement protocols and fault tolerance mechanisms in distributed systems.
- Describe the features of peer-to-peer and distributed shared memory systems

TEXT BOOKS: 1. Kshemkalyani, Ajay D., and Mukesh Singhal. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011.

2. George Coulouris, Jean Dollimore and Tim Kindberg, —Distributed Systems Concepts and Design®, Fifth Edition, Pearson Education, 2012.

REFERENCES: 1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.

2. Mukesh Singhal and Niranjan G. Shivaratri. Advanced concepts in operating systems. McGraw-Hill, Inc., 1994.

3. Tanenbaum A.S., Van Steen M., —Distributed Systems: Principles and Paradigms®, Pearson Education, 2007.

4. Liu M.L., —Distributed Computing, Principles and Applications®, Pearson Education, 2004. 5. Nancy A Lynch, —Distributed Algorithms®, Morgan Kaufman Publishers, USA, 2003.

TABLE OF CONTENT

CS8603 – DISTRIBUTED SYSTEMS		
Unit No	SYLLABUS	Page No.
I	INTRODUCTION	4-11
II	MESSAGE ORDERING & SNAPSHOTS	12-19
III	DISTRIBUTED MUTEX & DEADLOCK	20-26
IV	RECOVERY & CONSENSUS	27-34
V	P2P & DISTRIBUTED SHARED MEMORY	35-40

SubjectCode:CS8603
Subject Name: DISTRIBUTED SYSTEMS

Year/Semester:I/02
Subject Handler: Mr.H.Shine

UNIT I INTRODUCTION	
PART A	
Q.No.	Questions
1.	<p>Define – DistributedSystem.[BTL1] A distributed system is a collection of independent computers that appears to its users as a single coherent system. A distributed system is one in which components located at networked communicate and coordinate their actions only by passing</p>
2	<p>List the advantages of distributed systems overcentralized systems.[BTL1]</p> <p>The advantages of distributed systems over centralized systems are</p> <ul style="list-style-type: none"> a) economics, b) speed, c) inherent distribution, d) reliability and e) Incremental growth.
3	<p>Mention the examples of distributedsystem.[BTL1]</p> <p>The examples of distributed system are</p> <ul style="list-style-type: none"> a) internet, b) intranets, c) Mobile and ubiquitous computing.

4	<p>Mention the challenges in distributed system.[BTL1]</p> <p>The challenges in distributed system are</p> <ul style="list-style-type: none"> a) Heterogeneity, b) Openness, c) Security, d) Scalability, e) Failure handling, f) Concurrency and g) Transparency.
5	<p>What are the advantages of Distributed Systems?[BTL1]</p> <p>The advantages of distributed systems are</p> <ul style="list-style-type: none"> a) Performance, b) Distribution, c) Reliability (fault tolerance), d) Incremental growth, e) Sharing of data/resources and f) Communication.
6	<p>What are the disadvantages of Distributed Systems?[BTL1]</p> <p>The disadvantages of distributed systems are</p> <ul style="list-style-type: none"> h) Difficulties of developing distributed software, i) Networking problems and j) Security problems.
7	<p>Why we need openness?[BTL1]</p> <p>The degree to which a computer system can be extended and re-implemented is openness.</p> <p>IEEE = Institute of Electrical and Electronic Engineers</p> <p>ex: IEEE 802.11 WLAN, IEEE 802.3 Ethernet, W3C = World Wide Web Consortium</p>

8	<p>What is failure handling in distributedsystems? [BTL1]</p> <p>Any process, computer or network may fail independently of the others. Therefore each component needs to be aware of the possible ways .</p>
9	<p>What is the architectural model? [BTL1]</p> <p>An architectural model defines the way in which the components of system interact with one another and the way in which they are mapped onto an underlying network of computers.</p>
10	<p>What is the fundamental model? [BTL1]</p> <p>Fundamental models that help to reveal key problems for the designers of distributed systems. Their purpose is to specify the design issues, difficulties and threats that must be resolved in order to develop distributed systems that fulfill their tasks correctly, reliably and securely. The fundamental mode provides abstract views of just those characteristics of distributed systems that affect the dependability characteristics - correctness, reliability and security.</p>
11	<p>What is meant by inter-process communication? [BTL1]</p> <p>Inter process communication is concerned with the communication between processes in a distributed system, both in its own right and as support for communication between distributed objects. The Java API for inter process communication in the internet provides both datagram and stream communication.</p>
12	<p>What is marshalling and unmarshalling? [BTL1]</p> <p>Marshalling is the process of taking a collection of data items and assembling them into a form suitable for transmission in a message. Unmarshalling is the process of disassembling them on arrival to produce an equivalent collection of data items at the destination.</p>
13	<p>What is QOS in distributed systems? [BTL1]</p> <p>It is not sufficient to provide access to services in distributed systems. In particular, it is also important to provide guarantees regarding the qualities associated with such service access. Examples of such qualities include parameters related to performance, security and reliability.</p>
14	<p>What are the issues relating to datagram communication? [BTL1]</p> <ul style="list-style-type: none"> i. Message size ii. Blocking iii. Time out iv. Receive anyway
15	<p>What is the use of middleware? [BTL1]</p> <p>Middleware is a layer of software whose purpose is to mask heterogeneity and to provide a convenient programming model to application programmers. Middleware is represented by processes or objects in a set of computers that interact with each other to implement communication and resource sharing support for distributed applications.</p>

16	<p>What is meant by group communication? [BTL1]</p> <p>Group communication is a multicast operation is more appropriate- this is an operation that sends a single message from one process to each of the members of a group of process, usually in such a way that the membership of the group is transparent to the sender.</p>
17	<p>Name five reasons why to build distributed system.? [BTL1]</p> <p>The five reasons to build distributed system are</p> <ul style="list-style-type: none"> a) Resource sharing, b) Reliability, c) Communication, d) Computation speed up and e) Incremental growth.
18	<p>What do you mean by transparency? [BTL1]</p> <p>Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that system is perceived as a whole rather than as a collection of independent components.</p>
19	<p>What are the different forms of transparency? [BTL1]</p> <p>Eight forms of transparency are</p> <ul style="list-style-type: none"> a) Access transparency, b) Location transparency, c) Concurrency transparency, d) Replication transparency, e) Failure transparency, f) Mobility transparency, g) Performance transparency and h) Scaling transparency.
20	<p>What are the three components of security? [BTL1]</p> <p>Security for information resources has three components:</p> <ul style="list-style-type: none"> ✓ Confidentiality: protection against disclosure to unauthorized individuals. ✓ Integrity: protection against corruption. ✓ Availability: protection against interference with the means to access the resources.
21	<p>What is the use of firewall? [BTL1]</p> <p>A firewall can be used to form a barrier around an intranet to protect it from outside users but does not deal with ensuring the appropriate use of resources by users within the intranet.</p>
22	<p>What are the failures detected in DS? [BTL1]</p> <p>Masking failures: Some detected failures can be hidden or made less severe. Examples of hiding failures:</p> <ol style="list-style-type: none"> 1. Messages can be retransmitted when they fail to arrive 2. File data can be written to a pair of disks that if one is corrupted, the other may still be correct. <p>Tolerating failures:</p> <p>Most of the services in the Internet do exhibit failures. It would not be practical for them to detect and hide all the failures occur in such network. Their clients are designed to tolerate failures, which generally involve the users in that.</p>

	Recovery from failures: involves the design of software so that the state permanent data can be rolled back after a server has crashed.
23	<p>.What is synchronous DS? [BTL1]</p> <p>1) The time to execute each step of a process has known lower and upper bounds. 2) Each message transmitted over a channel is received within a known bounded time. 3) Each process has a local clock whose drift rate from real time has a known bound. 4) It is possible to suggest likely upper and lower bounds for process execution time, message delay and clock drift rates in a distributed system, but it is difficult to arrive at realistic values and to provide guarantees of the chosen values. 5) In a synchronous system it is possible to use timeouts, for example to detect the failure of a process.</p>
24	<p>What is asynchronous DS? [BTL1]</p> <p>1. Many distributed systems, such as the Intranet, qualify as asynchronous system. 2. An asynchronous distributed system is one in which there are no bounds on:</p> <p>1. Process execution speeds-for example, one process step may take only a picoseconds and another a century; all that can be said is that each step may take an arbitrarily long time. 2. Message transmission delays-for example, one message from process A to process B may be delivered in negligible time and another may take several years. In other words, a message may be received after an arbitrarily long time. 3. Clock drift rates- again, the drift rate of a clock is arbitrary.</p>
25	<p>List the main types of architectural model? [BTL1]</p> <p>a. Software architecture. b. System architecture.</p> <ul style="list-style-type: none"> ✓ Client server model ✓ Services provided by multiple servers. ✓ Proxy servers and cache. ✓ Peer processes.

PART * B

	<p>1. Explain the need of Distributed systems its characteristics with example ? [BTL2]</p> <ul style="list-style-type: none"> ➢ Fault tolerance ➢ Consistency ➢ Security ➢ Reliability ➢ Concurrent transactions
1	

2	<p>Explain the challenges to be considered in the design of DS?</p> <ul style="list-style-type: none"> ➤ Heterogeneity ➤ Openness ➤ Security ➤ Scalability ➤ Failure handling ➤ Concurrency ➤ Transparency ➤ Quality of service
3	<p>Describe the capabilities and rules for implementation of logical clocks. (BTL3)(13M)</p> <ul style="list-style-type: none"> • Lamport's Distributed Mutual Exclusion Algorithm is a permission based algorithm proposed by Lamport as an illustration of his synchronization scheme for distributed systems. In permission based timestamp is used to order critical section requests and to resolve any conflict between requests. • In Lamport's Algorithm critical section requests are executed in the increasing order of timestamps i.e a request with smaller timestamp will be given permission to execute critical section first than a request with larger timestamp. • Three type of messages (REQUEST, REPLY and RELEASE) are used and communication channels are assumed to follow FIFO order. • A site send a REQUEST message to all other site to get their permission to enter critical section. • A site send a REPLY message to requesting site to give its permission to enter the critical section. • A site send a RELEASE message to all other site upon exiting the critical section. • Every site S_i, keeps a queue to store critical section requests ordered by their timestamps. request_queue_i denotes the queue of site S_i • A timestamp is given to each critical section request using Lamport's logical clock. • Timestamp is used to determine priority of critical section requests. Smaller timestamp gets high priority over larger timestamp. The execution of critical section request is always in the order of their timestamp.

	Identify and explain the basic properties of scalar time. (BTL 3)	(13 M)
6	<ul style="list-style-type: none"> ➤ Proposed by Lamport in 1978 as an attempt to totally order events in a distributed system. Time domain is the set of non-negative integers. ➤ The logical local clock of a process p_i and its local view of the global time are squashed into one integer variable C_i. Rules R1 and R2 to update the clocks are as follows: R1: Before executing an event (send, receive, or internal), process p_i executes the following: <ul style="list-style-type: none"> ▪ $C_i := C_i + d$ ($d > 0$) ➤ In general, every time R1 is executed, d can have a different value; however, typically d is kept at 1. 	
7	List and explain the basic properties of vector time.(BTL1)	(13)
	<ul style="list-style-type: none"> ➤ The timestamp of an event is the value of the vector clock of its process when the event is executed. ➤ Figure shows an example of vector clocks progress with the increment value $d=1$. ➤ Initially, a vector clock is $[0, 0, 0, \dots, 0]$. ➤ The following relations are defined to compare two vector timestamps, <ul style="list-style-type: none"> ➤ vh and vk: $vh = vk \Leftrightarrow \forall x : vh[x] = vk[x]$ $vh \leq vk \Leftrightarrow \forall x : vh[x] \leq vk[x]$ $vh < vk \Leftrightarrow vh \leq vk \text{ and } \exists x : vh[x] < vk[x]$ ➤ vh and vk: $vh = vk \Leftrightarrow \forall x : vh[x] = vk[x]$ $vh \leq vk \Leftrightarrow \forall x : vh[x] \leq vk[x]$ $vh < vk \Leftrightarrow vh \leq vk \text{ and } \exists x : vh[x] < vk[x]$ ➤ If the process at which an event occurred is known, the test to compare two timestamps can be simplified as follows: If events x and y respectively occurred at processes p_i and p_j and are assigned timestamps vh and vk, respectively, <ul style="list-style-type: none"> ➤ then $x \rightarrow y \Leftrightarrow vh[i] \leq vk[i] \text{ and } y \Leftrightarrow vh[i] > vk[i] \wedge vh[j] < vk[j]$ 	

Summarize NTP for synchronizing system of physical clocks in distributed systems. [BTL2] (13M)

Christian's Algorithm :

n Observations

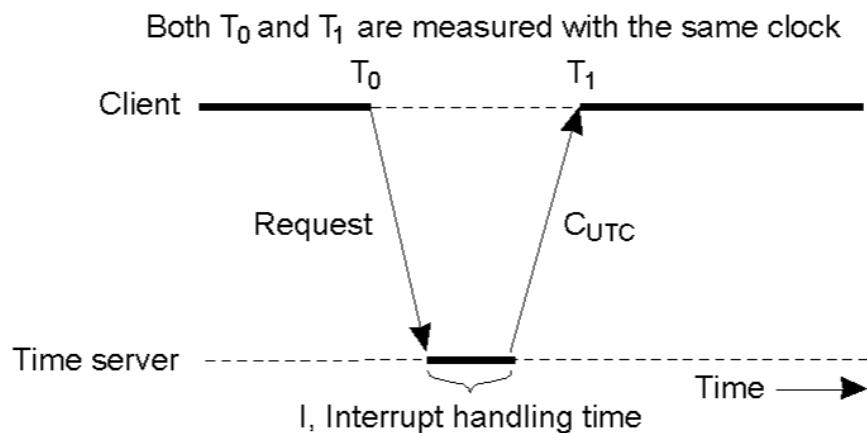
- n Round trip times between processes are often reasonably short in practice, yet theoretically unbounded n

Practical estimate possible if round-trip times are sufficiently short in comparison to required accuracy

n Principle:

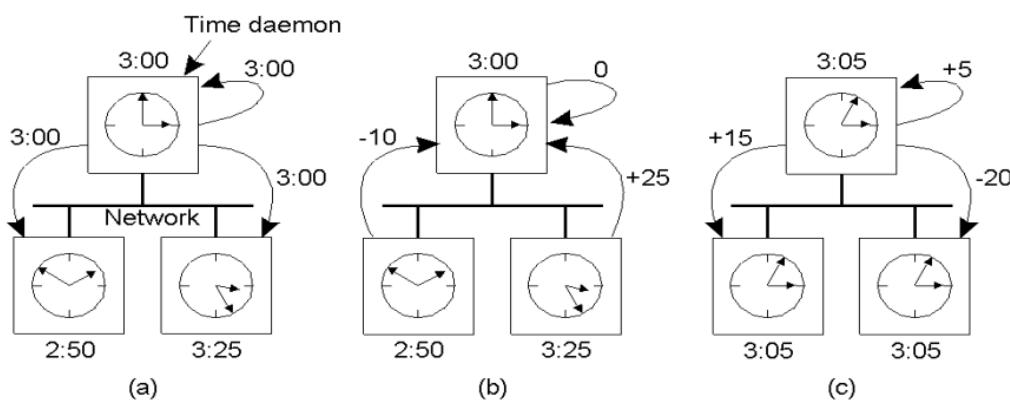
- n Use UTC-synchronized time server S
- n Process P sends requests to S n Measures round-trip time Tround - In LAN, Tround should be around 1-10 ms - During this time, a clock with a 10-6 sec/sec drift rate varies by at most 10-8 sec - Hence the estimate of Tround is reasonably accurate n

Naive estimate: Set clock to $t + \frac{1}{2}T_{\text{round}}$



The Berkeley Algorithm :

- The time daemon asks all the other machines for their clock values
- The machines answer
- The time daemon tells everyone how to adjust their clock





2106-JIT

UNIT II - MESSAGE ORDERING & SNAPSHOT	
Message ordering and group communication: Message ordering paradigms –Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system –Group communication – Causal order (CO) - Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions –Snapshot algorithms for FIFO channels .	
PART * A	
Q.No.	Questions
1.	What are the benefits of programming with interface in DS? [BTL1] <ul style="list-style-type: none"> I. As with any form of modular programming, programmers are concerned only with the abstraction offered by the service interface and need not be aware of implementation details. II. In potentially heterogeneous distributed systems , programmers also do not need to know the programming language or underlying platform used to implement the service. III. This approach provides natural support for software evolution in this implementation can change as long as the interface remains the same.
2	Define IDL. [BTL1] Interface Definition Languages (IDLs) are designed to allow procedures implemented in different languages to invoke one another. An IDL provides a notation for defining interfaces in which each of the parameters of an operation may be described as for input or output in addition to having its type specified.
3	List the used of IDL in web services. [BTL1] The concept of an IDL was initially developed for RPC systems but applies equally to RMI and also web service. Some of them are: I. Sun XDR as an example of an IDL for RPC II. CORBA IDL as an example of an IDL for RMI III. The web service Description Language (WSDL),which is designed for an Internet wide RPC supporting web service.
4	Define object reference. [BTL1] Objects can be accessed via object reference. For example in java a variable that appears to hold an object actually holds a reference to that object. To invoke a method in an object the object reference and method name are given together with any necessary arguments. The object whose method name is invoked is sometimes called the target and sometimes the receiver. Object reference are first class values, meaning that they may be assigned to variables, passed as arguments and returned as results of methods.
5	What is event notification? [BTL1] The distributed event based system extend the local event model by allowing multiple objects at different location to be notified of events takes place at an object. They use the publish subscribe paradigm. A publish subscribe system is a system where publishers publish structured events to an event service and subscriber express interest in particular events through subscriptions which can be arbitrary patterns over the structure events.

6	List the example of publish subscribe system.[BTL1] Publish-subscribe system is used in a wide variety of application domains particularly those related to a large scale dissemination of events. • Financial information systems. • Other area with live feeds of real time data(including RSS feeds)
7	Define callbacks. [BTL1] The general idea behind callbacks is that instead of clients polling the server to find out whether some event has occurred, the server should inform its clients whenever that event occurs. The term callback is used to refer to a server's action of notifying clients about an event.
8	Define process. [BTL1] A process consists of an execution environment together with one or more threads.
9	Define thread.[BTL1] A thread is the operating system abstraction of an activity (the term derives from the phrase 'thread of execution'). An execution environment is the unit of resource management: a collection of local kernel managed resources to which its threads have access.
10	Compare process and threads.[BTL1] a. Creation a new thread within an existing process is cheaper than creating a process. b. More importantly switching to a different thread within the same process is cheaper than switching between threads belonging to different processes. c. Threads within a process may share data and other resources conveniently and efficiently compared with separate processes. d. But by the same token threads within processes are not protected from one another.
11	Explain thread lifetime.[BTL1] A new thread is created on the same Java Virtual machine (JVM) as its creator in the SUSPENDED state. After it is made RUNNABLE with the start() method, it execute in the run() method of an object designated in its constructor, The JVM and the threads on top of it all execute in a process on top of the underlying operating system. Threads can be assigned a priority so that a java implementation that supports priorities will run a particular threads in preference to any thread with lower.
12	Define Global State. [BTL1] ➤ The global state of a distributed system is a collection of the local states of the processes and the channels. ➤ Notationally, global state GS is defined as, GS = { S _i L _{S<i>i</i>} , S _{i,j} C _{i,j} }
13	List the Issues in recording a global state. [BTL1] The following two issues need to be addressed: ➤ How to distinguish between the messages to be recorded in the snapshot from those not to be recorded. ➤ How to determine the instant when a process takes its snapshot.
	What is persistent object store?[BTL1] ➤ An object that is guaranteed to live between activations of processes is called a persistent object. Persistent objects are generally managed by persistent object stores, which store their state in a marshaled form on disk.

14	<ul style="list-style-type: none"> ➤ In generally a persistent object store will manage very large numbers of persistent object which are stored in a disk or in a database until they are needed.
15	<p>What is Multicast operation? [BTL1] A multicast operation is more appropriate – this is an operation that sends a single message from one process to each of the members of a group of processes, usually in such a way that the membership of the group is transparent to the sender.</p>
16	<p>What infrastructure provided by multicast message for distributed system?[BTL1]</p> <ul style="list-style-type: none"> ✓ Fault tolerance based on replicated services ✓ Discovering services in spontaneous networking ✓ o Better performance through replicated data o Propagation of event notifications.
17	<p>What is meant by client server communication?[BTL1] The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.</p>
18	<p>What are the three types of communication paradigm in distributed system?[BTL1] The three types of communication paradigm are:</p> <ul style="list-style-type: none"> ➤ Inter-process communication ➤ Remote invocation ➤ Indirect communication.
19	<p>Difference between synchronous and asynchronous communication?[BTL2]</p> <ul style="list-style-type: none"> ➤ In synchronous form of communication, the sending and receiving processes synchronize at every message. In this case, both send and receive are blocking operations. Whenever a send is issued the sending process is blocked until the corresponding receive is issued. Whenever receive is issued, the process blocks until a message arrives. ➤ In asynchronous form of communication, the use of the send operation is non-blocking in that the sending process is allowed to proceed as soon as the message has been copied to a local buffer and the transmission of the message proceeds in parallel with the sending process. The receive operation can have blocking and non-blocking variants.
20	<p>What is a Message queue?[BTL1] Message queues offer a point-to-point service whereby producer processes can send messages to a specified queue and consumer processes can receive messages from the queue or be notified of the arrival of new messages in the queue. Queues therefore offer an indirection between the producer and consumer processes.</p>
21	<p>What is meant by group communication?[BTL1] Group communication is a multicast operation is more appropriate- this is an operation that sends a single message from one process to each of the members of a group of process, usually in such a way that the membership of the group is transparent to the sender.</p>
22	<p>What is meant by inter process Communication?[BTL1] Inter process communication is concerned with the communication between processes in a distributed system, both in its own right and as support for communication between distributed objects. The Java API for inter process communication in the internet provides both datagram and stream</p>

23	<p>What is the difference between RMI and RPC? [BTL1]</p> <p>Remote Procedure Call or the RPC and the Remote Method Invocation or RMI are both message passing techniques in the Inter Process Communication (IPC).</p> <p>But there are two basic differences between the two methods:</p> <ol style="list-style-type: none"> 1. RPC supports procedural programming. i.e. only remote procedures can be invoked. Whereas RMI is object-based. As the name suggests, it is invoked on remote objects. 2. In RPC, the parameters that are passed are ordinary data structures. Whereas in RMI, objects can be passed as parameters.
24	<p>What is reliable communication?[BTL1]</p> <p>The term reliable communication is defined in terms of validity and integrity as follows:</p> <p>Validity: Any message in the outgoing message buffer is eventually delivered to the incoming message buffer.</p> <p>Integrity: The message received is identical to one sent, and no messages are delivered twice.</p>
25	<p>Define clock drift rate.[BTL1]</p> <p>The term clock drift rate refers to the rate at which a computer clock deviates from a perfect reference clock.</p>

PART * B	
1.	<p>1. Explain in detail about the architectural model of distributed system.? [BTL2]</p> <ul style="list-style-type: none"> • The architecture of a system is its structure in terms of separately specified components and their interrelationships. The overall goal is to ensure that the structure will meet present and likely future demands on it. <ul style="list-style-type: none"> ✓ Examples of architecture models are: ✓ Client – server model ✓ Peer to peer model ✓ Proxy server model ✓ Architectural elements: ✓ Communicating entities: <p>The first two questions above are absolutely central to an understanding of distributed systems; what is communicating and how those entities communicate together define a rich design space for the distributed systems developer to consider. It is helpful to address the first question from a system-oriented and a problem-oriented perspective.</p> <ul style="list-style-type: none"> • Communication paradigms: We now turn our attention to how entities communicate in a distributed system, and consider three types of communication paradigm: <ul style="list-style-type: none"> ○ interprocess communication; ○ remote invocation; ○ indirect communication. <ul style="list-style-type: none"> • Architectural patterns Layering The concept of layering is a familiar one and is closely related to abstraction. In a layered approach, a complex system is partitioned into a number of layers, with a given layer making use of the services offered by the layer below.

2	<p>Discuss about UDP datagram communication? [BTL2]</p> <ol style="list-style-type: none"> 1. UDP datagram communication A datagram sent by UDP is transmitted from a sending process to a receiving process without acknowledgement or retries. 2. If a failure occurs, the message may not arrive. A datagram is transmitted between processes when one process sends it and another receives it . 3. The following are some issues relating to datagram communication: Message size Blocking Timeouts Receive from any Failure model for UDP datagrams. 4. Omission failures Ordering: Use of UDP For some applications, it is acceptable to use a service that is liable to occasional omission failures. For example, the Domain Name System, which looks up DNS names in the Internet, is implemented over UDP. Voice over IP (VOIP) also runs over UDP. UDP datagrams are sometimes an attractive choice because they do not suffer from the overheads associated with guaranteed message delivery. 5. There are three main sources of overhead: <ul style="list-style-type: none"> ✓ the need to store state information at the source and destination; ✓ the transmission of extra messages; ✓ latency for the sender
3	<p>Illustrate IP multicast communication.? [BTL2]</p> <p>A multicast operation is more appropriate – this is an operation that sends a single message from one process to each of the members of a group of processes, usually in such a way that the membership of the group is transparent to the sender.</p> <p>Multicast messages provide a useful infrastructure for constructing distributed systems with the following characteristics:</p> <ol style="list-style-type: none"> 1. Fault tolerance based on replicated services 2. Discovering services in spontaneous networking 3. Better performance through replicated data 4. Propagation of event notifications <p>IP multicast – An implementation of multicast communication When a multicast message arrives at a computer, copies are forwarded to all of the local sockets that have joined the specified multicast address and are bound to the specified port number.</p> <p>The following details are specific to IPv4:</p> <ul style="list-style-type: none"> • Multicast routers • Multicast address

4

Explain in details about Snapshot algorithms for FIFO channels?[BTL2]**Chandy-Lamport algorithm :**

- ✓ The Chandy-Lamport algorithm uses a control message, called a marker whose role in a FIFO system is to separate messages in the channels.
- ✓ After a site has recorded its snapshot, it sends a marker, along all of its outgoing channels before sending out any more messages.
- ✓ A marker separates the messages in the channel into those to be included in the snapshot from those not to be recorded in the snapshot.
- ✓ A process must record its snapshot no later than when it receives a marker on any of its incoming channels.

5	<p>Explain in details about system model.? [BTL2]</p> <ul style="list-style-type: none"> ➤ The system consists of a collection of n processes p1, p2, ..., pn that are connected by channels. ➤ There are no globally shared memory and physical global clock and processes communicate by passing messages through communication channels. ➤ Cij denotes the channel from process pi to process pj and its state is denoted by SCij . ➤ The actions performed by a process are modeled as three types of events: ➤ Internal events, the message send event and the message receive event. ➤ For a message mij that is sent by process pi to process pj , let send(mij) and rec(mij) denote its send and receive events.
6	<p>List the responsibilities of core OS.? [BTL2]</p> <ul style="list-style-type: none"> ➤ The core OS components and their responsibilities are : ➤ Process manager: Creation of and operations upon process. ➤ A process is a unit of resource management, including an address space and one or more threads. ➤ Thread manager: Thread creation, synchronization and scheduling. ➤ Threads are schedulable activities attached to processes. ➤ Communication manager: Communication between threads attached to different processes on the same computer some kernels also support communication between threads in remote processes. Other kernels have no notion of other computers built into them, and an additional service is required for external communication. ➤ Memory manager: Management of physical and virtual memory. It describes the utilization of memory management techniques for efficient data copying and sharing. ➤ Supervisor: Dispatching of interrupts, system call traps and other exceptions; control of memory management unit and hardware caches; processor and floating-point unit register manipulation. This is known as the Hardware Abstraction Layer in Windows.

7	<p>Explain the Bagrodia's Algorithm for Binary Rendezvous ?[BTL2]</p> <ul style="list-style-type: none"> ➤ Assumptions Receives are always enabled Send, once enabled, remains enabled To break deadlock, PIDs used to introduce asymmetry Each process schedules one send at a time Message types: M, ack(M),request(M), permission(M) Process blocks when it knows it can successfully synchronize the current message P M ack(M) permission(M) request(M) M (a) (b) higher priority lower priority j P i ➤ Request(M) arrival at Pi from a lower priority process Pj : At the time a request(M) is processed by Pi , process Pi executes send(permission(M)) to Pj and blocks waiting for the message M. When M arrives, the RECEIVE(M) event is executed and the process unblocks. ➤ Message M arrival at Pi from a higher priority process Pj : At the time a message M is processed by Pi , process Pi executes RECEIVE(M) (which is assumed to be always enabled) and then send(ack(M)) to Pj . ➤ Processing when Pi is unblocked: When Pi is unblocked, it dequeues the next (if any) message from the queue and processes it as a message arrival (as per Rules 3 or 4)
---	---

UNIT III – DISTRIBUTED MUTEX & DEADLOCK	
Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport’s algorithm – Ricart-Agrawala algorithm – Maekawa’s algorithm – Suzuki-Kasami’s broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries – Models of deadlocks – Knapp’s classification – Algorithms for the single resource model, the AND model and the OR model.	
PART * A	
Q.No	Questions
1	<p>Define Mutual exclusion. [BTL1]</p> <ul style="list-style-type: none"> ✓ Concurrent access of processes to a shared resource or data is executed in mutually exclusive manner. ✓ Only one process is allowed to execute the critical section (CS) at any given time. ✓ In a distributed system, shared variables (semaphores) or a local kernel cannot be used to implement mutual exclusion.
2	<p>What are the basic approaches for distributed mutual exclusion? [BTL1]</p> <p>Three basic approaches for distributed mutual exclusion:</p> <ol style="list-style-type: none"> 1 Token based approach 2 Non-token based approach 3 Quorum based approach
3	<p>What is meant by Token-based approach? [BTL1]</p> <ul style="list-style-type: none"> ❖ A unique token is shared among the sites. ❖ A site is allowed to enter its CS if it possesses the token. ❖ Mutual exclusion is ensured because the token is unique.

4	<p>Define Quorum based approach. [BTL1]</p> <ul style="list-style-type: none"> ✓ Each site requests permission to execute the CS from a subset of sites (called a quorum). ✓ Any two quorums contain a common site. ✓ This common site is responsible to make sure that only one request executes the CS at any time.
5	<p>What is goal of peer to peer system?[BTL1]</p> <p>The goal of peer-to-peer systems is to enable the sharing of data and resources on a very large scale by eliminating any requirement for separately managed servers and their associated infrastructure.</p>
6	<p>What are the characteristics of peer to peer system? [BTL1]MAY/JUNE 2016</p> <p>Their design ensures that each user contributes resources to the system.</p> <ul style="list-style-type: none"> • Although they may differ in the resources that they contribute, all the nodes in a peer-to-peer system have the same functional capabilities and responsibilities. • Their correct operation does not depend on the existence of any centrally administered systems. • They can be designed to offer a limited degree of anonymity to the providers.
7	<p>What is napster file system? [BTL1]NOV/DEC 2018</p> <p>The need for and the feasibility of a peer-to-peer solution were first demonstrated by the Napster file sharing system [Open Nap 2001] which provided a means for users to share files. Napster became very popular for music exchange. At its peak, several million users were registered and thousands were swapping music files simultaneously.</p>
8	<p>What is the need of peer to peer middleware system? [BTL1]</p> <p>Peer-to-peer middleware systems are designed specifically to meet the need for the automatic placement and subsequent location of the distributed objects managed by peer-to-peer systems and applications</p>
9	<p>Write the Non-functional requirements of peer-to-peer middleware system?[BTL1]</p> <ul style="list-style-type: none"> • Global scalability • Load balancing ➤ Optimization for local interactions between neighbouring peers o Accommodating to highly dynamic host availability.
11	<p>What are the tasks performed by routing overlay?[BTL1]</p> <ul style="list-style-type: none"> • Insertion of objects • Deletion of objects • Node addition and removal.
12	<p>What are the case studies used in overlay?[BTL1] NOV/DEC 2017</p> <ul style="list-style-type: none"> • Pastry is the message routing infrastructure deployed in several applications including PAST.

- | | |
|--|---|
| | <ul style="list-style-type: none">o Tapestry is the basis for the Ocean Store storage system. |
|--|---|

2106-JIT

	What is meant by distributed file system?[BTL1]
13	A distributed file system enables programs to store and access remote files exactly as they do local ones, allowing users to access files from any computer on a network.
14	What are the two basic file system used in distributed system?[BTL1] <ul style="list-style-type: none"> • The Sun Network File System, NFS. • The Andrew File System, AFS.
15	Define metadata.[BTL1] The term metadata is often used to refer to all of the extra information stored by a file system that is needed for the management of files. It includes file attributes, directories and all the other persistent information used by the file system.
16	What are the different forms of transparency are partially or wholly addressed by current file services?[BTL1] <ul style="list-style-type: none"> • Access transparency • Location transparency • Mobility transparency • Performance transparency • Scaling transparency.
17	Define File service architecture of AFS. [BTL1] An architecture that offers a clear separation of the main concerns in providing access to files is obtained by structuring the file service as three components – <ul style="list-style-type: none"> ✓ a flat file service, ✓ a directory service and ✓ a client module.
18	What is Andrew File System?[BTL1] Andrew is a distributed computing environment developed at Carnegie Mellon University (CMU) for use as a campus computing and information system. The design of the Andrew File System (henceforth abbreviated AFS) reflects an intention to support information sharing on a large scale by minimizing client-server communication.

	Write the Key design issues for distributed file systems?[BTL1]
19	The Key design issues for distributed file systems are: <ul style="list-style-type: none"> ○ the effective use of client caching to achieve performance equal to or better than that of local file systems; ○ the maintenance of consistency between multiple cached client copies of files when they are updated; • recovery after client or server failure; ➤ high throughput for reading and writing files of all sizes; Scalability
20	What is the need of Name service?[BTL1] In a distributed system, names are used to refer to a wide variety of resources such as computers, services, remote objects and files, as well as to users. A name is needed to request a computer system to act upon a specific resource chosen out of many.
21	Define Identifier. [BTL1] Identifier is sometimes used to refer to names that are interpreted only by programs. Remote object references and NFS file handles are examples of identifiers. Identifiers are chosen for the efficiency with which they can be looked up and stored by software.
22	What is caching?[BTL1] Caching • In DNS and other name services, client name resolution software and servers maintain a cache of the results of previous name resolutions. When a client requests a name lookup, the name resolution software consults its cache. If it holds a recent result from a previous lookup for the name, it returns it to the client; otherwise, it sets about finding it from a server. That server, in turn, may return data cached from other servers.
23	Discuss on LDAP.[BTL1]MAY /JUNE 2016 LDAP (Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.
24	Demonstrate the use of name cache[BTL1] ✓ To increase the performance of name service ✓ To reduce overall system overhead .
25	Define Gnutella.[BTL1]NOV/DEC 2018 Gnutella (possibly by analogy with the GNU Project) is a large peer-to-peer network. It was the first decentralized peer-to-peer network of its kind, leading to other, later networks adopting the model.
	Part * B
1	Explain the details of Agrawala algorithm achieves mutual exclusion.? [BTL2] <ul style="list-style-type: none"> ✓ Theorem: Ricart-Agrawala algorithm achieves mutual exclusion. ✓ Proof: Proof is by contradiction. Suppose two sites Si and Sj' are executing the CS concurrently and Si 's request has higher priority than the request of Sj .

	<ul style="list-style-type: none"> ✓ Clearly, S_i received S_j's request after it has made its own request. Thus, S_j can concurrently execute the CS with S_i only if S_i returns a REPLY to S_j (in response to S_j's request) before S_i exits the CS. ✓ However, this is impossible because S_j's request has lower priority. Therefore, Ricart-Agrawala algorithm achieves mutual exclusion.
	<ul style="list-style-type: none"> ✓ For each CS execution, Ricart-Agrawala algorithm requires $(N - 1)$ REQUEST messages and $(N - 1)$ REPLY messages. Thus, it requires $2(N - 1)$ messages per CS execution. Synchronization delay in the algorithm is T.
2	<p>Explain the details about Singhal's Dynamic Information-Structure Algorithm? [BTL2]</p> <ul style="list-style-type: none"> ✓ Most mutual exclusion algorithms use a static approach to invoke mutual exclusion. ✓ These algorithms always take the same course of actions to invoke mutual exclusion no matter what is the state of the system. ✓ These algorithms lack efficiency because they fail to exploit the changing conditions in the system. ✓ An algorithm can exploit dynamic conditions of the system to improve the performance.
	<p>Explain in details about the Quorum-Based Mutual Exclusion?[BTL2]</p> <ul style="list-style-type: none"> ➤ Algorithms Quorum-based mutual exclusion algorithms are different in the following two ways: 1 A site does not request permission from all other sites, but only from a subset of the sites. ➤ The request set of sites are chosen such that $\forall i \forall j :$ ➤ $1 \leq i, j \leq N :: R_i \cap R_j = \emptyset$. Consequently, every pair of sites has a site which mediates conflicts between that pair. ➤ 2 A site can send out only one REPLY message at any time. A site can send a REPLY message only after it has received a RELEASE message for the previous REPLY message.
3	<p>Explain in details about the Algorithm for constructing a tree-structured quorum? [BTL2]</p> <ul style="list-style-type: none"> • The algorithm tries to construct quorums in a way that each quorum represents any path from the root to a leaf. • If it fails to find such a path (say, because node 'x' has failed), the control goes to the ELSE block which specifies that the failed node 'x' is substituted by two paths both of which start with the left and right children of 'x' and end at leaf nodes. • If the leaf site is down or inaccessible due to any reason, then the quorum cannot be formed and the algorithm terminates with an error condition. • The sets that are constructed using this algorithm are termed as tree quorums.

Explain the Suzuki-Kasami's Broadcast Algorithm? [BTL2]

If a site wants to enter the CS and it does not have the token, it broadcasts a REQUEST message for the token to all other sites. A site which possesses the token sends it to the requesting site upon the receipt of its REQUEST message. If a site receives a REQUEST message when it is executing the CS, it sends the token only after it has completed the execution of the CS.

This algorithm must efficiently address the following two design issues:

- How to distinguish an outdated REQUEST message from a current REQUEST message:

Due to variable message delays, a site may receive a token request message after the corresponding request has been satisfied.

- If a site can notdetermined if the request corresponding to a token request has been satisfied, it may dispatch the token to a site that does not need it.
- This will not violate the correctness, however, this may seriously degrade the performance.

(2) How to determine which site has an outstanding request for the CS: After a site has finished the execution of the CS, it must determine what sites have an outstanding request for the CS so that the token can be dispatched to one of them.

:

Explain in details about Raymond's Tree-Based Algorithm? [BTL2]

5

- ✓ This algorithm uses a spanning tree to reduce the number of messages exchanged per critical section execution. The network is viewed as a graph, a spanning tree of a network is a tree that contains all the N nodes.
- ✓ The algorithm assumes that the underlying network guarantees message delivery. All nodes of the network are 'completely reliable'.
- ✓ The algorithm operates on a minimal spanning tree of the network topology or a logical structure imposed on the network.
- ✓ The algorithm assumes the network nodes to be arranged in an unrooted tree structure. Figure 4 shows a spanning tree of seven nodes A, B, C, D, E, F, and G.
- ✓ Messages between nodes traverse along the undirected edges of the tree.
- ✓ A node needs to hold information about and communicate only to its immediate-neighboring nodes. Similar to the concept of tokens used in token-based algorithms, this algorithm uses a concept of privilege.
- ✓ Only one node can be in possession of the privilege (called the privileged node) at any time, except when the privilege is in transit from one node to another in the form of a PRIVILEGE message.
- ✓ When there are no nodes requesting for the privilege, it remains in possession of the node that last used it.

	<p>Explain in details about Distributed deadlock detection? [BTL2]</p> <p>An algorithm for detecting deadlocks in a distributed system was proposed by Chandy, Misra, and Haas in 1983. Processes request resources from the current holder of that resource. Some processes may wait for resources, which may be held either locally or remotely. Cross-machine arcs make looking for cycles, and hence detecting deadlock, difficult. This algorithm avoids the problem of constructing a Global WFG.</p> <p>6 The Chandy-Misra-Haas algorithm works this way: when a process has to wait for a resource, a probe message is sent to the process holding that resource. The probe message contains three components: the process ID that blocked, the process ID that is sending the request, and the destination. Initially, the first two components will be the same. When a process receives the probe: if the process itself is waiting on a resource, it updates the sending and destination fields of the message and forwards it to the resource holder.</p> <p>If it is waiting on multiple resources, a message is sent to each process holding the resources. This process continues as long as processes are waiting for resources. If the originator gets a message and sees its own process number in the blocked field of the message, it knows that a cycle has been taken and deadlock exists. In this case, some process (transaction) will have to die. The sender may choose to commit suicide and abort itself or an election algorithm may be used to determine an alternate victim (e.g., youngest process, oldest process, ...).</p>
--	---

	UNIT – IV RECOVERY & CONSENSUS
	Checkpointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated checkpointing algorithm – Algorithm for asynchronous checkpointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure – free system – Agreement in synchronous systems with failures.
	PART * A
1	<p>What is clock skew and clock drift? [BTL1]APRIL/MAY 2018</p> <p>The instantaneous difference between the readings of any two clocks is called their skew. Clock drift means that they count time at different rates, and so diverge.</p>

2	What is clocks drift rate?[BTL1] A clock's drift rate is the change in the offset (difference in reading) between the clock and a nominal perfect reference clock per unit of time measured by the reference clock.
3	What is meant by International Atomic Time ?[BTL1] Computer clocks can be synchronized to external sources of highly accurate time. The most accurate physical clocks use atomic oscillators, whose drift rate is about one part in 10^{13} . The output of these atomic clocks is used as the standard for elapsed real time, Known as International Atomic Time.
4	What is Coordinated Universal Time?[BTL1] Coordinated Universal Time – abbreviated as UTC (from the French equivalent) – is an international standard for timekeeping. It is based on atomic time, but a so-called ‘leap second’ is inserted – or, more rarely, deleted – occasionally to keep it in step with astronomical time. UTC signals are synchronized and broadcast regularly from landbased radio stations and satellites covering many parts of the world.
5	How the clock synchronization done in Cristian's method?[BTL1] A single time server might fail, so they suggest the use of a group of synchronized servers . It does not deal with faulty .
6	Explain the BerKley algorithm.[BTL1] The BerKeley algorithm eliminates readings from faulty clocks. Such clocks could have a significant adverse effect if an ordinary average was taken so instead the master takes a faulttolerant average. That is, a subset is chosen of clocks that do not differ from one another by more than a specified amount, and the average is taken of readings from only these clocks.
7	What are the issues resolved by BerKley's algorithm?[BTL1] The collection of computers whose clocks are to be synchronized are categorized as masters and slaves. The averaging of the clock values cancels out the individuals clocks tendencies to run fast or slow. This overcomes the uncertainty due to message transmission time introduced in the Synchronized clock values returned by the master.
8	What is network partition?[BTL1] The network partition can be used to separate a group of replica managers into two or more sub groups. The members of same subgroup communicate with one another but members of different subgroup cannot communicate with one another.
9	Write the features of Network Time Protocol?[BTL1] To provide a service enabling clients across the Internet to be synchronized accurately to UTC To provide a reliable service that can survive lengthy losses of connectivity: To enable clients to resynchronize sufficiently frequently to offset the rates of drift found in most computers: To provide protection against interference with the time service, whether malicious or accidental:

	Define election algorithm. [BTL1]
10	<ul style="list-style-type: none"> ✓ Mention the different algorithm? An algorithm for choosing a unique process to play a particular role is called an election algorithm. Ex: In a variant of central server algorithm for mutual exclusion, the server is chosen from among the process . ✓ The different algorithms are <ul style="list-style-type: none"> • Ring based election algorithm • Bully algorithm.
12	Explain global states and consistent cuts with example. ?[BTL1] Global state of a distributed system consists of –Local state of each process: messages sent and messages received –State of each channel: messages sent but not received.
13	Explain the bully algorithm? [BTL1]NOV/DEC 2017 This algorithm allows process to crash during an election. Although it assumes that message delivery between process is reliable. It assumes that the system is synchronous – it uses timeouts to detect a process failure.
14	What is the use of Multicast operation?[BTL1] It is generally useful to allow processes to be members of several groups simultaneously – for example, to enable processes to receive information from several sources by joining several.
15	What is internal synchronization?[BTL1] And if the clocks C are synchronized with one another to known degree of accuracy, then we can measure the interval between two events occurring at different computers by appealing to their local clocks, even though they are not necessarily synchronized to an external source of time. This is internal synchronization. <ul style="list-style-type: none"> ♣ For a synchronization bound $D > 0$ and for a source S of UTC times, $S(t) - Ci(t) < D$, for all real times t in I. ♣ Clocks C agree with in the bound D
16	What is strata? [BTL1] The NTP service is provided by a network of servers located across the Internet. Primary servers are connected directly to a time source such as a radio clock receiving UTC; secondary servers are synchronized, ultimately, with primary servers. The servers are connected in a logical hierarchy called a synchronization subnet whose levels are called strata.
17	What is filter dispersion? [BTL1] NTP servers apply a data filtering algorithm to successive pairs which estimates the offset o and calculates the quality of this estimates as a statistical quantity called the filter dispersion
18	What is synchronization dispersion? [BTL1] Peers with lower stratum numbers are more favoured than those in higher strata because they are ‘closer’ to the primary time sources. Also, those with the lowest synchronization dispersion are relatively favoured. This is the sum of the filter dispersions measured between the server and the root of the synchronization subnet.
19	What is list cloning? BTL1 List cloning is a process of copying data of one list to another list. There are two types of cloning Deep copy and shallow copy
20	What is list aliasing? BTL1 In list aliasing, items of one list will be copied to other list. Change in one list will affect the other

21	What do you meant by distributed garbage ?BTL1 An object is considered to be garbage if there are no longer any reference to it anywhere in the distributed system. The memory taken up by that object can be reclaimed once it is known as to be garbage.
22	Define Global History BTL1 Let us return to our general system p of N processes $p_i (i=1,2,3,\dots,N)$. Here a series of events occurs at each process, and that we may characterize the execution of each process by its history.
23	Define Failure detector.BTL1 A failure detector is a service that processes queries about whether a particular process has failed. It is often implemented by an object local to each process that runs failure detection algorithms in conjunction with its counterparts at the other processes.
24	What is meant by election Election? BTL1 choosing a unique process for a particular role is called an election – All the processes agree on the unique choice – For example, server in dist. mutex
25	List the famous mutual exclusion algorithms. BTL1 <ul style="list-style-type: none"> • Center server algorithm • Ring- Based algorithms • Mutual Exclusion using multicast and Logical Clocks
	PART * B
1	Explain the two phase commit protocol with an example [BTL2] (13M) During the progress of a transaction, there is no communication between the coordinator and the participants apart from the participants informing the coordinator when they join the transaction. A client's request to commit (or abort) a transaction is directed to the coordinator. If the client requests abortTransaction, or if the transaction is aborted by one of the participants, the coordinator informs all participants immediately. It is when the client asks the coordinator to commit the transaction that the two-phase commit protocol comes into use. The two-phase commit protocol Phase 1 (voting phase): 1. The coordinator sends a canCommit? request to each of the participants in the transaction. 2. When a participant receives a canCommit? request it replies with its vote (Yes or No) to the coordinator. Before voting Yes, it prepares to commit by saving objects in permanent storage. If the vote is No, the participant aborts immediately. Phase 2 (completion according to outcome of vote): 3. The coordinator collects the votes (including its own). (a) If there are no failures and all the votes are Yes, the coordinator decides to commit the transaction and sends a doCommit request to each of the participants. (b) Otherwise, the coordinator decides to abort the transaction and sends doAbort requests to all

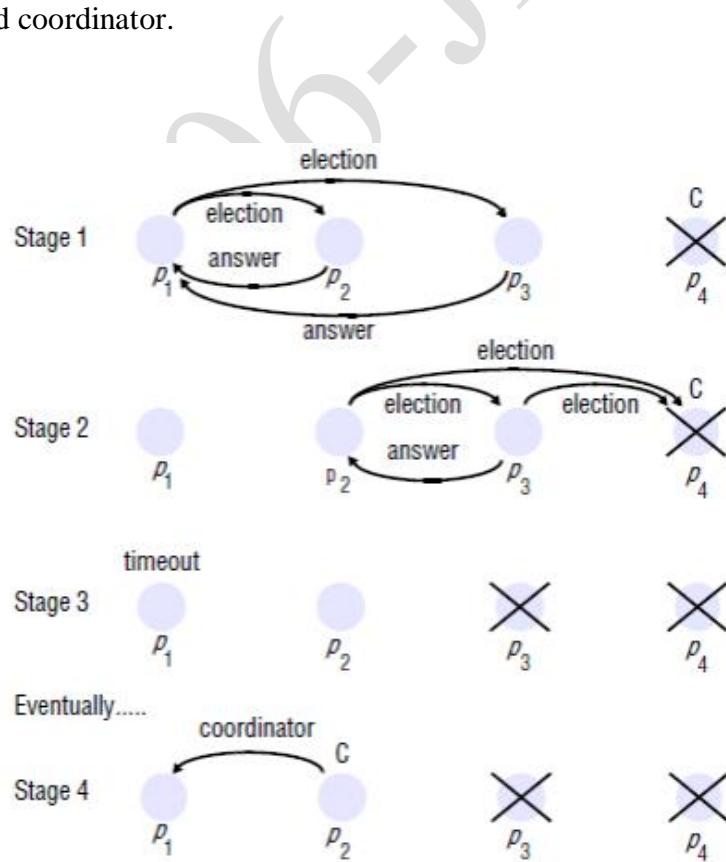
participants that voted Yes.

4. Participants that voted Yes are waiting for a doCommit or doAbort request from the coordinator. When a participant receives one of these messages it acts accordingly and, in the case of commit, makes a haveCommitted call as confirmation to the coordinator

Explain bully's algorithms? [BTL 2]

The bully algorithm allows processes to crash during an election, although it assumes that message delivery between processes is reliable. Unlike the ring-based algorithm, this algorithm assumes that the system is synchronous: it uses timeouts to detect a process failure.

- 2 The operation of the algorithm is shown in above figure. There are four processes, p1 – p4 . Process p1 detects the failure of the coordinator p4 and announces an election (stage 1 in the figure). On receiving an election message from p1 , processes p2 and p3 send answer messages to p1 and begin their own elections; p3 sends an answer message to p2 , but p3 receives no answer message from the failed process p4 (stage 2). It therefore decides that it is the coordinator. But before it can send out the coordinator message, it too fails (stage 3). When p1 's timeout period T_{exp} expires (which we assume occurs before p2 's timeout expires), it deduces the absence of a coordinator message and begins another election. Eventually, p2 is elected coordinator.



3 Explain global states and consistent cuts in details? [BTL2]

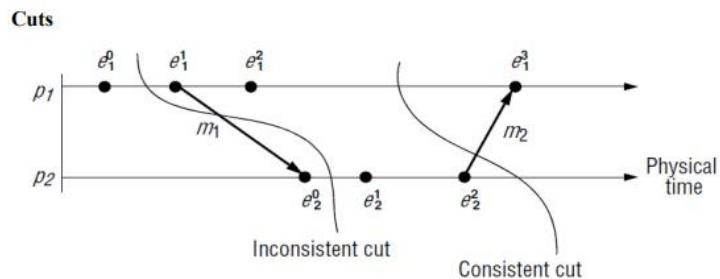
It is possible in principle to observe the succession of states of an individual process, but the question of how to ascertain a global state of the system

- the state of the collection of processes
- is much harder to address.

The essential problem is the absence of global time.

If all processes had perfectly synchronized clocks, then we could agree on a time at which each process would record its state – the result would be an actual global state of the system.

From the collection of process states we could tell, for example, whether the processes were deadlocked. But we cannot achieve perfect clock synchronization, so this method is not available to us.



UNIT-V

P2P & DISTRIBUTED SHARED MEMORY

Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry. Distributed shared memory: Abstraction and advantages – Memory consistency models – Shared memory Mutual Exclusion.

Q.No	PART * A
1	What is process? [BTL 1] Process means a program in execution. Process execution must progress sequential order.
2	What is process migration? [BTL 1] The phenomenon of shifting a process from one machine to another one which is called process migration.
3	What is Load? [BTL 1] Load may be define as number of tasks are running in queue, CPU utilization, load average, I/O utilization, amount of free CPU time/memory, etc.

4	<p>List desirable features of good process migration mechanism.[BTL 1]</p> <ul style="list-style-type: none"> • Transparency • Efficiency • Minimal interference • Minimize freezing time • Minimal residual dependencies
5	<p>List any three challenges of process migration. [BTL 1]</p> <ul style="list-style-type: none"> • Process state capturing and transfer • Scheduling • System call
6	<p>What are strategies for the migration of files?[BTL 1]</p> <ul style="list-style-type: none"> • If the file is locked by the migrating process and resides on the same system, then transfer file with the process • If the process is moved temporarily, transfer the file only after an access request was made by the migrated process.
7	<p>Define thread.[BTL 1]</p> <p>A minimal software processor in which context a series of instructions can be executed. Saving a thread context implies stopping the current execution and saving all the data needed to continue the execution at a later stage</p>
8	<p>Explain the benefit of process migration. [BTL 1]</p> <ul style="list-style-type: none"> • Better response time and execution speed – up • Reducing network traffic • Improving system reliability • Higher throughput and effective resource utilization
9	<p>List the types of process scheduling techniques.[BTL 1]</p> <ul style="list-style-type: none"> • Task management approaching • Load balancing approaching • Load – Sharing approaching
10	<p>What is kernel level thread? [BTL 1]</p> <p>In kernel level thread, thread management is done by kernel. OS support the kernel level thread. Since kernel managing threads, kernel can schedule another thread if a given thread blocks rather than blocking the entire processes.</p>
11	<p>What is user level thread? [BTL 1]</p> <p>User level thread uses user space for thread scheduling. These threads are transparent to the operating system. User level threads are created by runtime libraries that cannot execute privileged instructions.</p>
12	<p>What is preemptive process migration?[BTL 1]</p> <p>Preemptive process transfer involve the transfer of a process that is partially executed. This transfer is an expensive operation as the collection of a process's state can be difficult.</p>
13	<p>What is non preemptive process migration?[BTL 1]</p> <p>Non –preemptive process transfers involve the transfer of process that have not begun execution and hence do not require the transfer of the process state. In both types of transfers, information about the environment in which the process will execute must be transferred to the receiving node.</p>

14	<p>Explain distributed shared memory. [BTL2]</p> <p>Distributed shared memory(DSM) system is a resource management component of distributed operating system that implements shared memory model in distributed system which have no physically shared memory. The shared memory model provides a virtual address space which is shared by all nodes in a distributed system.</p>
15	<p>Discuss the three properties of weak consistency[BTL 2]</p> <p>It is impossible for a distributed computer system to simultaneously provide Consistency, Availability and Partition Tolerance. A distributed system can satisfy any two of these guarantees at the same time but not all three. N1 and N2 are networks which both share a piece of data v.</p>
16	<p>What is entry consistency? [BTL 1]</p> <p>This is a variant of the release consistency model. It also requires the use of acquire and release instructions to explicitly state an entry or exit to a critical section. However, under entry consistency, every shared variable is assigned a synchronization variable specific to it.</p>
17	<p>What are the two steps involved in chord protocol? [BTL 1]</p> <p>The application interacts with Chord in two main ways. First, the Chord library provides a lookup(key) function that yields the IP address of the node responsible for the key. Second, the Chord software on each node notifies the application of changes in the set of keys that the node is responsible for.</p>
18	<p>Define Namespaces. BTL3</p> <p>Variables are names or identifiers that map to objects. A namespace is a dictionary of variable names/keys and their corresponding objects values. Each function has its own local namespace.</p>
19	<p>Mention the attributes related to file object. BTL3</p> <ul style="list-style-type: none"> ✓ File.closed ✓ file.mode ✓ file.name ✓ file.softspace
20	<p>Discuss content addressable networks (CAN)? [BTL 6]</p> <p>The Content Addressable Network (CAN) is a distributed, decentralized P2P infrastructure that provides hash table functionality on an Internet-like scale. CAN was one of the original four distributed hash table proposals, introduced concurrently with Chord, Pastry, and Tapestry.</p>
21	<p>What is Try and Except? BTL1</p> <p>If an error is encountered, a try block code execution is stopped and transferred down to the except block.</p> <p>In addition to using an except block after the try block, you can also use the finally block. The code in the finally block will be executed regardless of whether an exception occurs.</p>

22	<p>How to raise an exception? BTL1</p> <p>You can raise an exception in your own program by using the raise exception [, value] statement.</p> <p>Raising an exception breaks current code execution and returns the exception back until it is handled.</p>
23	<p>List some exception error. BTL1</p> <ul style="list-style-type: none"> ✓ IOError ✓ ImportError ✓ ValueError ✓ KeyboardInterrupt
24	<p>What is command line argument? BTL1</p> <p>Command line arguments are values passed in during execution of a program. These values are passed after the file name.</p>
25	<p>What are the performance features of P2P systems? [BTL 1]</p> <p>Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.</p>
PART * B	
1	<p>Explain in detail about process migration.[BTL 2]</p> <p>Process migration is the transfer of a sufficient amount of the state of a process from one computer to another for the process to execute on the target machine.</p> <p>Working of process migration:</p> <ul style="list-style-type: none"> • Selecting a process to be migrated • Selecting the destination node • Suspending the process • Capturing the process state • Sending the state to the destination • Resuming the process • Forwarding future messages to the destination <p>Process migration is of two types:</p> <ul style="list-style-type: none"> • Preemptive process migration • Non – preemptive process migration <p>Desirable features of good process migration mechanism</p> <ul style="list-style-type: none"> • Transparency • Minimal interference • Minimal residual dependencies • Efficiency • Robustness Process migration mechanisms • Freezing and restarting a process • Transferring the address space and state • Message – forwarding mechanism Advantages of process migration • Balancing the load • Moving the process closer to the resources it is using utilizes resources effectively and reduces

	<p>network traffic</p> <ul style="list-style-type: none"> • Being able to move a copy of a process on another node improves system reliability
2	<p>Explain in detail about thread? [BTL 2]</p> <p>Thread is a dispatchable unit of work. It consists of thread ID, program counter, stack and register set.</p> <p>Thread advantages</p> <ul style="list-style-type: none"> • Context switching time is minimized • Thread support for efficient communication • Resources sharing is possible using threads • A thread provides concurrency within a process
3	<p>Illustrate thread model. [BTL3]</p> <ul style="list-style-type: none"> • User level thread • Kernel level thread <p>User level thread User level thread uses user space for thread scheduling. These threads are transparent to the operating system. User level threads are created by runtime libraries that cannot execute privileged instructions.</p> <p>Advantage:</p> <ul style="list-style-type: none"> • Kernel mode privilege does not require for thread switching • These threads are fast to create and manage • User level threads are more portable <p>Disadvantage:</p> <ul style="list-style-type: none"> • Not suitable for multiprocessor system • User level threads also do not support system wide scheduling priority Kernel level thread In kernel level thread, thread management is done by kernel. OS support the kernel level thread. <p>Advantages:</p> <ul style="list-style-type: none"> • Each thread can be thread separately • Kernel routines itself as multithreaded. <p>Disadvantage:</p> <ul style="list-style-type: none"> • Slower than user level thread. • There will be overhead and increased in kernel complexity
4	<p>Explain in detail about resource management [BTL2]</p> <p>Distributed systems contain a set of resources interconnected by a network. The processes are migrated to fulfill their resource requirements and resource manager are to control the assignment of resources to processes.</p> <p>Types of process scheduling techniques</p> <ul style="list-style-type: none"> • Task assignment approach • Load balancing approach • Load sharing approach <p>Features</p> <ul style="list-style-type: none"> • Dynamic in nature • Quick decision making capability • Balanced system performance

	<ul style="list-style-type: none"> • Stability • Fault tolerance • Fairness of service
5	<p>Explain in detail about Task assignment approach [BTL 2]</p> <ul style="list-style-type: none"> • Process have been split into tasks • Computational requirement of tasks and speed of processors are known • Costs of processing tasks on nodes are known • Reassignment of task is not possible <p>Goals of task assignment algorithm</p> <ul style="list-style-type: none"> • Minimization of IPC costs • Quick turnaround time for the complete process • A high degree of parallelism • Efficient utilization of system resources <p>Task assignment algorithms are of three types:</p> <ul style="list-style-type: none"> • Graph theoretic deterministic algorithm • Centralized heuristics algorithm • Hierarchical algorithm
6	<p>Examine the chord protocol with simple key lookup algorithm. [BTL 4]</p> <p>A fundamental problem that confronts peer-to-peer applications is the efficient location of the node that stores a desired data item. This paper presents Chord, a distributed lookup protocol that addresses this problem.</p> <p>Chord provides support for just one operation: given a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key/data pair at the node to which the key maps.</p> <p>Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is continuously changing.</p> <p>Results from theoretical analysis and simulations show that Chord is scalable: Communication cost and the state maintained by each node scale logarithmically with the number of Chord nodes.</p>

IT8076**SOFTWARE TESTING****LTPC
3003****UNIT I INTRODUCTION****9**

Testing as an Engineering Activity – Testing as a Process – Testing Maturity Model- Testing axioms – Basic definitions – Software Testing Principles – The Tester’s Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design –Defect Examples- Developer/Tester Support of Developing a Defect Repository.

UNIT II TEST CASE DESIGN STRATEGIES**9**

Test case Design Strategies – Using Black Box Approach to Test Case Design – Boundary Value Analysis – Equivalence Class Partitioning – State based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing – Random Testing – Requirements based testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Additional White box testing approaches- Evaluating Test Adequacy Criteria.

UNIT III LEVELS OF TESTING**9**

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing –Compatibility testing – Testing the documentation – Website testing.

UNIT IV TEST MANAGEMENT**9**

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group- The Structure of Testing Group- . The Technical Training Program.

UNIT V TEST AUTOMATION**9**

Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.

Subject Code: IT8076 Year/Semester :III/06**Subject Name : Software Testing****Subject Handler: Ms.R.Revathi****UNIT I INTRODUCTION**

Testing as an Engineering Activity – Testing as a Process – Testing axioms – Basic definitions – Software Testing Principles – The Testers Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer/Tester Support of Developing a Defect Repository

PART* A

Q.NO	QUESTIONS
1.	Define Software Engineering.(BTL1) Software Engineering is a discipline that produces error free software with in a time and budget.
2.	Define software Testing.(AU April/May 2016)(BTL1) Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.
3.	List the elements of the engineering disciplines.(BTL1) <ul style="list-style-type: none"> • Basic principles • Processes • Standards • Measurements • Tools • Methods • Best practices • Code of ethics • Body of knowledge
4.	Define process in the context of software quality.(BTL1) “Process” in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals
5.	Define the term Testing(BTL1) <ul style="list-style-type: none"> • Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software. • Testing can be described as a process used for revealing defects in software, and for

	establishing that the software has attained a specified degree of quality with respect to selected attributes.
6.	<p>Interpret the term Debugging or fault localization.(BTL1)</p> <p>Debugging or fault localization is the process of</p> <ul style="list-style-type: none"> • Locating the fault or defect • Repairing the code, and • Retesting the code.
7.	<p>List the levels of TMM.(AU Nov/Dec2016)(BTL1)</p> <p>The testing maturity model or TMM contains five levels.</p> <p>They are</p> <ul style="list-style-type: none"> • Level1: Initial • Level2: Phase definition • Level3: Integration • Level4: Management and Measurement • Level5: Optimization /Defect prevention and Quality Control.
8.	<p>List the members of the critical groups in a testing process. (BTL1)</p> <ul style="list-style-type: none"> • Manager • Developer/Tester • User/Client
9.	<p>Define Error.(BTL1)</p> <p>An error is mistake or misconception or misunderstanding on the part of a software developer.</p>
10.	<p>Define Faults (Defects).(AU Nov/Dec2016) (BTL1)</p> <p>A fault is introduced into the software as the result of an error. It is an anomaly in the software that may cause it to behave incorrectly, and not according to its specification.</p>
11.	<p>Define failures.(AU NOV/DEC 2016) (BTL5)</p> <p>A failure is the inability of a software or component to perform its required functions within specified performance requirements.</p>
12	<p>What is the need of TMM?</p> <p>Test maturity model gives the level at which an organization stands in meeting Testing criterias.</p>
13	<p>Define Validation.(BTL1)</p> <p>Validation is the process of evaluating a software system or component during, or at the end of, the development cycle in order to determine whether it satisfies specified requirements.</p>
14	<p>Explain in short about Verification.(BTL1)</p> <p>Verification is the process of evaluating a software system or component to determine whether the product of a given development phase satisfy the conditions imposed at the start of that phase.</p>
15	<p>Programmer A and Programmer B are working on a group of interfacing modules. Programmer A tends to be a poor communicator and does not get along well with Programmer B. Due to this situation, what types of defects are likely to surface in these interfacing modules?(BTL3)</p> <ul style="list-style-type: none"> • Communication defects.

16	List the stages in Software Engineering (BTL1) <ul style="list-style-type: none"> • Requirement Analysis • Design • Coding • Deployment • Delivery
17	Define Test Cases. (BTL1) A test case in a practical sense is a test related item which contains the following information. A set of test inputs. These are data items received from an external source by the code under test. The external source can be hardware, software, or human. Execution conditions. These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device. Expected outputs. These are the specified results to be produced by the code under test.
18.	Define Test Oracle (BTL1) Test Oracle is a document, or a piece of software that allows tester to determine whether a test has been passed or failed.
19.	Define Test Bed.(AU Nov/Dec 2017)(BTL1) A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.
20.	Define Software Quality. (BTL1) Quality relates to the degree to which a system, system component, or process meets specified requirements. Quality relates to the degree to which a system, system component, or process meets Customer or user needs, or expectations.
21.	List the Quality Attributes.(BTL1) <ul style="list-style-type: none"> • Correctness • Reliability • Usability • Integrity • Portability • Maintainability • Interoperability
22.	Define SQA group.(BTL1) The software quality assurance (SQA) group is a team of people with the necessary training and skills to ensure that all necessary actions are taken during the development process so that the resulting software confirms to established technical requirements.
23.	Explain the work of SQA group.(BTL2) Testers to develop quality related policies and quality assurance plans for each project. The group is also involved in measurement collection and analysis, record keeping, and Reporting. The SQA team members participate in reviews and audits, record and track Problems, and verify that corrections have been made.
24.	Define reviews. (BTL1) A review is a group meeting whose purpose is to evaluate a software artifact or a set of Software artifacts. Review and audit is usually conducted by a SQA group.
25.	List the sources of Defects or Origins of defects. (AU April/May 2017)(BTL1) <ul style="list-style-type: none"> • Education • Communication

- Oversight
- Transcription
- Process.

PART *B

1	Discuss about the role of process in software quality (Testing). (13M) BTL2
---	--

Answer: page : 1 - Notes

Process – Creates an impact in the system.(2M)

Testing as a process: Set of activities well planned in advance.(2M)

Figure - Components of an engineered process.(2M) (Pg no: 2 in notes)

Explanation:

Testing Maturity model TMM (4M)

- Level 1: Initial
- Level 2: Phase Definition
- Level 3: Integration
- Level 4: Management and Measurement
- Level 5: Optimization/Defect Prevention/Quality control

Verification and Validation Model(3M)

- **Diagram :** Pg no:2 in notes
- **Verification:** Checks if software confirm to Functional and Non – Functional requirements
- **Validation:** Confirms if Software meets user requirements.

2	Draw the 5-level structure of the testing maturity model ,discuss about it.(13M) BTL2
---	--

Answer : page : 9 - Notes

Test maturity model : Gives an overview of the activities done in each level of testing.

Explanation: Testing Maturity model TMM (4M)

- Level 1: Initial
- Level 2: Phase Definition
- Level 3: Integration
- Level 4: Management and Measurement
- Level 5: Optimization/Defect Prevention/Quality control

Verification and Validation Model(3M)

- **Diagram :** Pg no:2 in notes (4M)
- **Verification:** Checks if software confirm to Functional and Non – Functional requirements
- **Validation:** Confirms if Software meets user requirements.

3	Explain in detail about the software testing principles.(13M)
---	--

April/may 2017BTL2

Answer : Page: 3-23 - Srinivasan & Ramaswamy

Definition: Principle is any rule that governs the system.

Explanation:

- Principle 1 :Revealing defects and evaluating quality (2M)
- Principle 2 :Effectiveness of testing effort (1M)
- Principle 3 : Test results should be inspected (1M)
- Principle 4 : Test case must contain the expected output (1M)
- Principle 5:Test case developed for both valid and invalid input conditions (1M)

	<ul style="list-style-type: none"> • Principle 6 :Defects ratio (1M) • Principle7 : Testing should be carried out by a group (1M) • Principle8 : Tests must be repeatable and reusable (1M) • Principle9 :Testing should be planned (1M) • Principle 10: Testing activities should be integrated into software lifecycle (1M) • Principle 11: Testing is a creative and challenging task (2)
4	<p>Give an example for defect classes and discuss them in detail.(13M) (Nov/Dec 2016) BTL4</p> <p>Answer : page : 10 - Notes</p> <p>Definition: Any abnormal condition that affects the execution of a program is called defect. (2M)</p> <p>Figure: Defect classes and a defect repository.(2M)</p> <p>Explanation: (9M)</p> <ul style="list-style-type: none"> • Requirements and specification defects <ol style="list-style-type: none"> Functional Description defects Feature defects Feature interaction defects Interface description defects, • Design defects <ol style="list-style-type: none"> Algorithmic and processing defects Control ,logic, and sequence defects Data defects. Module interface description defects External Interface description defects. • Coding defects <ol style="list-style-type: none"> Algorithmic and processing defects. Control ,logic, and sequence defects Typographical defects Initialization defects. Dataflow defects Data defects Module interface defects, Code document defects. External hardware and software interface defects, Testing defects. Test harness defects Test case design and test procedure Defects.
5	<p>Explain in detail about Testing as a Process. (13M)BTL2</p> <p>Answer : Page :29 - 31 - Srinivasan & Ramaswamy book</p> <p>Process – Creates an impact in the system.(2M)</p> <p>Testing as a process: Set of activities well planned in advance.(2M)</p> <p>Figure - Components of an engineered process.(2M) (Pg no: 2 in notes)</p> <p>Explanation:</p> <p>Testing Maturity model TMM (4M)</p> <ul style="list-style-type: none"> • Level 1: Initial • Level 2: Phase Definition • Level 3: Integration • Level 4: Management and Measurement

	<ul style="list-style-type: none"> • Level 5: Optimization/Defect Prevention/Quality control <p>Verification and Validation Model(3M)</p> <ul style="list-style-type: none"> • Diagram : Pg no:2 in notes • Verification: Checks if software confirm to Functional and Non – Functional requirements <p>Validation: Confirms if Software meets user requirements.</p>
6	<p>Give a detailed account on the origins of defects. (13M) (Nov/Dec 2016) BTL2</p> <p>Answer : page: 10 - Notes</p> <p>Definition: Any abnormal condition that affects the execution of a program is called defect. (2M)</p> <p>Figure: Defect classes and a defect repository.(4M)</p> <p>Explanation: (3M)</p> <p>Types of Defect:</p> <ul style="list-style-type: none"> • Variance from product specification • Variance from customer or user specification • Wrong requirement • Missing Requirement • Extra Requirement <p>Sources of Error:</p> <ul style="list-style-type: none"> • Education • Communication • Oversight • Transcription • Process <p>Hypothesis:</p> <ul style="list-style-type: none"> • Design Test <p>Fault Model: Employed to prepare fault list.</p> <p>Physical Defects:</p> <ul style="list-style-type: none"> • Manufacturing Errors • Component Wear out. • Environmental Effects <p>Figure : Origins of defects.(4M)</p>
7	<p>Discuss the Coin Problem along with the defect categories associated withit.(13M) BTL6</p> <p>Answer : page : 17 - Notes</p> <p>Figure : Sample specification with defects.(4M)</p> <p>Figure : a sample design specification with defects(4M)</p> <p>Explanation:(5M)</p> <ul style="list-style-type: none"> • Algorithmic and processing defects. • Precondition • Post-condition • Control,logic, and sequence defects. • Typographical defects.

	<ul style="list-style-type: none"> Initialization defects. Dataflow defects.
8	<p>Analyse the role of tester in software development Organization.(13M) BTL4 (Nov/Dec 2017)</p> <p>Answer : page:9 - Notes</p> <p>Tester: Objective of testing to get high quality software which should satisfy all requirements of software. Role of tester to ensure whether all requirements of software are satisfied. (2M)</p> <p>Explanation:</p> <p>Tester's job:(4M)</p> <ul style="list-style-type: none"> Reveal defects Find weak points Inconsistent behavior Circumstances where the software does not work as expected. <p>Tester's Need:</p> <ul style="list-style-type: none"> Communication Skills Team working skills Decision Making skills <p>Testers are said to be specialist:</p> <ul style="list-style-type: none"> Provide plan Do Execution Recording result Analysing the test result
	PART – C
1	<p>Given 6 different denominations of coins ,The program finds total dollars & cents values for a set of coins and outputs the number of dollars. Find the possible defects in the above scenario.(15M) BTL6 (April/May 2017)</p> <p>Answer : Appendix - Srinivasan,Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> Requirements or functional Defects(4M) Functional description defects Interface description defects Pre conditions(5M) Post conditions Control,Logic and sequence defects(3M) Algorithmic and processing defects Data flow defects(3M) Data Defects External
2	<p>If you were testing a feature of your software on Monday and finding a new bug every hour, at what rate would you expect to find bugs on Tuesday? (15M) BTL4</p> <p>Answer : Appendix - Srinivasan,Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> Number of bugs remaining is proportional to the number of bugs you have already found.(8M) Pesticide paradox – Tells you that if you continue to run the same tests over and over that you eventually won't find new and different bugs until you add more tests.(7M)
3	Visiting all the states that the program has assures that you have also traverses all the

transitions among them. The statement is true or false? Justify your answer.(15M) BTL4

Answer : Appendix - Srinivasan Ramaswamy

STATEMENT :False(8M)

Explanation(7M)

- Think of visiting 50 different cities spread out across the entire United states.
- You could plan a trip that would take you to each city.
- But it would be impossible for you to travel all the roads that connects all the cities.

JIT - 2106

UNIT II TEST CASE DESIGN STRATEGIES

Test case Design Strategies – Using Black Box Approach to Test Case Design – Random Testing – Requirements based testing – Boundary Value Analysis – Equivalence Class Partitioning – State-based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Evaluating Test Adequacy Criteria.

PART *A

1	Define Smart Tester.(BTL1) Software must be tested before it is delivered to users. It is responsibility of the testers to Design tests that (i) reveal defects (ii) can be used to evaluate software performance, usability and reliability. To achieve these goals, tester must select a finite no. of test cases (i/p, o/p, & conditions).
2	Define responsibility.(AU Nov/Dec 2016)(BTL1) A straightforward definition for object-responsibility is this: An object must contain the data (attributes) and code (methods) necessary to perform any and all services that are required by the object.
3	Write short notes on Random testing and Equivalence class portioning.(BTL1) (Nov/Dec 2015) Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing. In equivalence class partitioning the input and output is divided in to equal classes or partitions.
4	Define State.(BTL1) A state is an internal configuration of a system or component. It is defined in terms of the values assumed at a particular time for the variables that characterize the system or component.
5	Define Finite-State machine. (BTL1) A finite-state machine is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.
6	Define Error Guessing.(BTL1) The tester/developer is sometimes able to make an educated “guess” as to which type of defects may be present and design test cases to reveal them. Error Guessing is an ad-hoc approach to test design in most cases.
7	Define COTS Components.(BTL1) The reusable component may come from a code reuse library within their org or, as is most likely, from an outside vendor who specializes in the development of specific types of software components. Components produced by vendor org are known as commercial off-the shelf, or COTS, components.
8	Express the benefits of low coupling.(BTL2) Maintainability – changes are confined in a single module Testability – modules involved in unit testing can be limited to a minimum Readability – classes that need to be analysed are kept at a minimum.
9	Define usage profiles and Certification.(BTL1) Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product,process, or service meets a specific set of requirements.

10	Write the application scope of adequacy criteria?(BTL4) <ul style="list-style-type: none"> Helping testers to select properties of a program to focus on during test. Helping testers to select a test data set for a program based on the selected properties. Supporting testers with the development of quantitative objectives for testing Indicating to testers whether or not testing can be stopped for that program.
11	Define path.(BTL1) A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.
12	Write the formula for cyclomatic complexity?(AU Nov/Dec 2016)(BTL1) The complexity value is usually calculated from control flow graph(G) by the formula. $V(G) = E - N + 2$ Where The value E is the number of edges in the control flow graph The value N is the number of nodes.
13	List the various iterations of Loop testing. ?(BTL1) <ul style="list-style-type: none"> Zero iteration of the loop One iteration of the loop Two iterations of the loop K iterations of the loop where $k < n$ $n-1$ iterations of the loop $n+1$ iterations of the loop
15	What are the errors uncovered by black box testing?(BTL1) <ul style="list-style-type: none"> Incorrect or missing functions Interface errors Errors in data structures Performance errors Initialization or termination error.
16	Define Equivalence class partitioning?(BTL1) If a tester is viewing the software-under-test as a black box with well defined inputs and outputs, a good approach to selecting test inputs is to use a method called Equivalence class partitioning.
17	Define Cause effect graphing?(BTL1) Cause Effect Graph is a black box testing technique that graphically illustrates the relationship between a given outcome and all the factors that influence the outcome.
18	What is Certification?(BTL1) Certification refers to third-party assurance that a product, process, or service meets a specific set of requirements.
19	What is the goal of smart tester?(BTL1) The goal of the smart tester is to understand the functionality, input/output domain, and the environment of use for the code being tested.
20	List the two major assumptions in Mutation testing.(BTL1) <ul style="list-style-type: none"> The component programmer hypothesis The coupling effects
21	List the two basic Testing strategies.(BTL1) <ul style="list-style-type: none"> Black box testing. White box testing.
22	What are the knowledge sources for Black box testing?(BTL2) <ul style="list-style-type: none"> Requirement Document specification Domain knowledge

	<ul style="list-style-type: none"> Defect analysis data
23	<p>What are the knowledge sources for White box testing? (AU Nov/Dec 2015)(BTL2)</p> <ul style="list-style-type: none"> High level design Detailed design Control flow graphs Cyclomatic complexity
24	<p>List the methods of Black box testing?(AU Nov/Dec 2017)(BTL1)</p> <ul style="list-style-type: none"> Equivalence class partitioning Boundary value analysis State transition testing Cause and effect graphing Error guessing
25	<p>List the methods of White box testing?(AU Nov/Dec 2017)(BTL1)</p> <ul style="list-style-type: none"> Statement testing Branch testing Path testing Data flow testing Mutation testing Loop testing
PART* B	
1	<p>Elaborate the qualities of a smart Tester.BTL2(13M)</p> <p>Answer : page : 18 - Notes</p> <ul style="list-style-type: none"> Reveal defects(4M) <ul style="list-style-type: none"> Find the bugs before the software becomes operational Find errors at the early stage (Requirement Analysis) Find the weak points Situations at which error may occur Evaluate quality(4M) <ul style="list-style-type: none"> Ensures if software meets user requirements Ensures if software meets requirement specification Ensures if software meets performance criteria such as reliability,usability, portability Finite no of test case(5M) <ul style="list-style-type: none"> Number of test cases. A test case that makes the tester to makesure that software meets all user requirements. Test cases that are capable enough to make the system to crash.
2	<p>Discuss the test case design strategies.BTL2 (13M)</p> <p>Answer: page : 18 - Notes</p> <ul style="list-style-type: none"> Two strategies Whitebox (clear or glass box) (2M) Testing the software with X – Ray glasses Black box(Functional or specification)(2M) Testing the software blind folded.

	<ul style="list-style-type: none"> Table: The two basic testing strategies.(9M)Page 18 in notes
3	<p>List and explain the types of black box testing.AU April/May 2016 BTL2 (13M)</p> <p>Answer:Page:73 - 105 - Srinivasan & Ramaswamy</p> <ul style="list-style-type: none"> Random testing (1M) <ul style="list-style-type: none"> Randomly select the input. Three conditions. Equivalence class partitioning(2M) <ul style="list-style-type: none"> Adv of Equivalence class partitioning List of conditions. Figure: A specification of a square root function Example of equivalence class reporting table Boundary value analysis(1M) <ul style="list-style-type: none"> List the conditions Figure: Boundaries of on Equivalence partition Example of Boundary value analysis. State Transition Testing(1M) <ul style="list-style-type: none"> Abstract Machine State graph having a finite number of states and transitions between Internal configuration of system or component Error guessing(1M) <ul style="list-style-type: none"> Tester/Developer's past experience Cause and Effect Graphing(2M) <ul style="list-style-type: none"> Nodes in the graph are causes and effects Tester need to identify causes and effects Graph must be annotated with constraints Graph is then converted into decision table Columns in the decision table are converted into test cases Requirement Based Testing(1M) <ul style="list-style-type: none"> Test Requirement Specification Explicit Requirement Implicit Requirement Requirement traceability Matrix Compatibility Testing(1M) <ul style="list-style-type: none"> Confirms working of product with different infrastructure components Forward Compatibility Testing Backward Compatibility testing User documentation Testing(2M) <ul style="list-style-type: none"> Manuals, User guidelines Installation guidelines Setup guidelines, Readme files Software Release notes,Online help Domain Testing(1M) <ul style="list-style-type: none"> Needs business domain knowledge than software knowledge They get trained in software ,instead of training the software professional in business domain.

4	<p>Discuss the various approaches in White Box test design.(13M)</p> <p>AU Nov/Dec 2016 BTL2</p> <p>Answer: Page :29 - 31 - Srinivasan & Ramaswamy</p> <ul style="list-style-type: none"> • Coverage and control flow graph(3M) <ol style="list-style-type: none"> 1. Three basic primes 2. Sequential 3. Condition 4. Iteration • Coverage code logic(3M) <ol style="list-style-type: none"> 1. Figure: Code sample with branch and loop. 2. Figure: A control flow graph representation for the code. 3. Table: A test case for the code ,that satisfies the decision • coverage criterion.(3M) <ol style="list-style-type: none"> 1. Table: Test cases for simple decision coverage 2. Table: Test cases for condition coverage 3. Table: Test cases for decision condition coverage. • Path Testing (4M) <ol style="list-style-type: none"> 1. Path 2. Cyclomatic complexity formula.
5	<p>Evaluate test adequacy Criteria with necessary properties.(13M) BTL3</p> <p>Answer: page:27 - Srinivasa & Ramamurty</p> <ul style="list-style-type: none"> • Axioms –Set of assumptions(1M) • Applicability Property(1M) • Non exhaustive applicability property(1M) • Monotonicity Property(2M) • Inadequate Empty set(1M) • General multiple change Property(1M) • Anti decomposition Property(2M) • Renaming Property(1M) • Complexity Property(1M) • Statement Coverage Property(2M)
6	<p>Demonstrate the various black box testing approaches using Equivalence partitioning and boundary value Analysis.(13M)</p> <p>Nov/Dec 2016 BTL5</p> <p>Answer: Page : 84,90 - Srinivasan & Ramaswamy</p> <p>Equivalence Partition: (8M)</p> <p>Software testing technique – divides input data of software unit into partitions of equivalent data – test cases can be derived – main principle of test cases are deigned to coverpartition at least once.</p> <p>Boundary value analysis: (5M)</p> <p>Test case design technique to test boundary value between partitions-boundary value is an input or output value on the border of an equivalence partition.</p>
7	<p>Compare static testing with that of dynamic testing and list the major difference between both.(13M) BTL4</p>

		STATIC TESTING	DYNAMIC TESTING
Prevention.	Cure.		
More cost-effective.	Less cost – effective.		
Greater marginal benefits.	Lesser marginal benefits.		
Comprehensive diagnostics for code.	More diagnostics for code.		
Finds more bugs.	Finds fewer bugs.		
Takes lesser time.	Takes longer time.		
Testing covers more areas.	Testing covers less areas.		
Done in verification stage.	done in validation stage.		

PART *C

1	Explain What a tester should worry about with this line from a spec. The software will allow up to 100 million simultaneous connections, although no more than 1 million will normally be used.BTL4(15 M) Answer : Appendix - Srinivasan Ramaswamy <ul style="list-style-type: none"> • Testability(4M) • It doesn't matter that typical usage is only 1 million connections.(4M) • If the specification states that 100 million are possible(4M) • The 100 million must be tested.(3M)
2	Assume that you are assigned to test the windows calculator, Is it possible to test all the test cases. How do you test it systematically and explain the principle involved.BTL6(15M) Answer: Appendix - Srinivasan Ramaswamy <ul style="list-style-type: none"> • Equivalence Partitioning(5M) • Grouping similar input(4M) • Grouping similar Output(4M) • Grouping similar operations of software(2M)
3	Visiting all the states that the program has assures that you have also traverses all the transitions among them. The statement is true or false? Justify your answer.BTL4(15M) Answer : Appendix - Srinivasan Ramaswamy False(8M) Explanation(7M) <ul style="list-style-type: none"> • Think of visiting 50 different cities spread out across the entire United States. • You could plan a trip that would take you to each city. • But it would be impossible for you to travel all the roads that connect all the cities.

UNIT III LEVELS OF TESTING

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing – Compatibility testing – Testing the documentation – Website testing.

PART A

1	Define Unit Testing (Nov/Dec 2017)BTL1 A unit is the smallest possible testable software component that can be characterized in several ways.
2	Write the different levels of testing.BTL1 <ul style="list-style-type: none"> • Unit test • Integration test • System test • Acceptance test.
3	List the components suitable for unit test.BTL1 <ul style="list-style-type: none"> • Procedures and functions • Classes/objects and methods Procedure-sized reusable components.
4	List the phases in the unit test planning. (April/May 2015)BTL1 <ul style="list-style-type: none"> • Phase 1: Describe unit test approach and risks. • Phase 2: Identify unit features to be tested. • Phase 3: Add levels of detailed to the plan.
5	Write the issues in the unit test.BTL1 <ul style="list-style-type: none"> • Issue 1: Adequately testing classes. • Issue 2: Observation of objects states and state changes. • Issue 3: The retesting of classes-I • Issue 4: The retesting of classes-II
6	What is Test harness?(Nov/Dec 2016)BTL1 The auxiliary code developed to support to testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.
7	List the major goals of Integration test.BTL1 <ul style="list-style-type: none"> • To detect defects that occurs on the interfaces of units. • To assemble the individual units into working subsystems and the finally a complete system that is ready for system test
8	What is the advantage of Bottom up integration?BTL1 Bottom-up integration has the advantage that the lower-level modules are usually well tested early in the integration process. This is important if these modules are candidates for reuse.
9	What is a cluster?BTL1 A cluster consists of classes that are related, for example, they may work together to support a required functionality for the complete system.

10	List the several types of system tests.(Nov/Dec 2016)BTL1 <ul style="list-style-type: none"> • Functional testing • Performance testing • Stress testing • Configuration testing • Security testing • Recovery testing
11	Define Load. BTL1 A load is a series of inputs that simulates a group of transactions.
12	List the two major requirements of Performance testing. BTL1 <ul style="list-style-type: none"> • Functional requirements • Quality requirements.
13	What is meant by Stress testing? BTL1 When a system is tested with a load that causes it to allocate its resources in maximum amounts, this is called stress testing.
14	Define Recovery testing. BTL1 Recovery testing subjects a system to losses of resources in order to determine if it can recover properly from these losses.
15	Define Use case. BTL1 A use case is a pattern, scenario, or exemplar of usage. It describes a typical interaction between the software system under development and a user.
16	Define Regression testing . BTL1 Regression testing is not a level of testing, but it is the retesting of the software that occurs when the changes are made to ensure that the new version of the software has retained the capabilities of the old version and that has no defect have been introduced due to the changes.
17	Write the objectives of configuration testing. BTL1 <ul style="list-style-type: none"> • Show that all the configuration changing commands and menus work properly • Show that all interchangeable devices are really interchangeable, and that they each enter the proper states for the specified conditions • Show that the system's performance level is maintained when devices are interchanged, or when they fail.
18	List the effect of security breaches. BTL1 <ul style="list-style-type: none"> • Loss of information • Corruption of information • Misinformation • Privacy violations • Denial of service
19	Define functional Testing. BTL1 Functional tests at the system level are used ensure that the behavior of the system adheres to the requirement specifications.
20	What is load generator and Load? BTL1 An important tool for implementing system tests is a load generator. A load generator is essential for testing quality requirements such as performance and stress. A load is a series of inputs that simulates a group of transactions.
21	What are the approaches used to develop the software? BTL1 There are two major approaches to software development

	<ul style="list-style-type: none"> • Bottom-Up • Top-Down
22	List the objectives of configuration testing. BTLL <ul style="list-style-type: none"> • Show that all the configuration changing commands and menus work properly • Show that all interchangeable devices are really interchangeable, and that they each enter the proper states for the specified conditions • Show that the system's performance level is maintained when devices are interchanged, or when they fail.
23	List the effect of security breaches. BTLL <ul style="list-style-type: none"> • Loss of information • Corruption of information • Misinformation • Privacy violations • Denial of service.
24	Give the examples of security testing. BTLL2 <ul style="list-style-type: none"> • Password checking. • Legal and illegal entry with password. • Password Expiration. • Encryption. • Browsing. • Trap doors. • Viruses.
25	List the areas covered during recovery testing. BTLL <ul style="list-style-type: none"> • Restart. • Switchover.

PART B

1	<p>How would you define a software unit? In terms of your definition, what constitutes a unit for procedural code; for object-oriented code?(13M)BTLL4</p> <p>Answer : Page : 38,261-264 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Functions, procedures, classes and methods as units • Fig: Some components suitable for unit test(1M) • Unit Test: Need for preparation <ul style="list-style-type: none"> • Planning • Both black box and White box • Reviewe • Several Tasks <p>1. Unit Test Planning(4M)</p> <p>Phase I: Describe unit test approach and Risks</p> <p>Phase II: Identify unit features to be tested</p> <p>Phase III: Add levels of detail to the planning</p> <p>2. Designing the Unit Test(3M)</p> <ul style="list-style-type: none"> • Test Cases • Test Procedure <p>3. Running and recording the results(3M)</p>
---	---

	<ul style="list-style-type: none"> • Perform the unit test in all the units of system • Record the results. <p>4. Test Harness(2M)</p> <ul style="list-style-type: none"> • Additional code included to perform testing.
2	<p>Why is it so important to design a test harness for reusability?(13M)BTL2</p> <p>Answer: Page : 35 - Notes</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Auxiliary code developed • Support testing of units , components (3M) • Harness consists of drivers that call the target code • Stubs that represent modules it calls.(3M) <p>Fig: The test Harness(4M)</p> <p>Driver(2M)</p> <p>Stub(1M)</p>
3	<p>What are the key differences in integrating procedural-oriented systems as compared to object-oriented systems?(13M)BTL3</p> <p>Answer: Page : 35 - Notes</p> <ul style="list-style-type: none"> • Goals(2M) • Integration Strategies: <ul style="list-style-type: none"> i. Top – Down ii. Bottom – Up iii. Bi – Directional • Designing Integration Test: <ul style="list-style-type: none"> i. Black Box Approach ii. White Box Approach • Integration test strategy for procedures(5M) • Integration test strategy for classes(6M) • Critical Module characteristics
4	<p>Describe the activities/Tasks and responsibilities for developer/testers in support of multilevel testing.(13M)BTL2</p> <p>Answer : Page :261 - Srinivasan & Ramaswamy</p> <ul style="list-style-type: none"> • Fig: Levels of testing <p>Explanation:</p> <ul style="list-style-type: none"> • Levels of Testing(4M) <ul style="list-style-type: none"> i. Unit Test ii. Integration test iii. System Test iv. Acceptance Test • Two Approaches(4M) <ul style="list-style-type: none"> Bottom_Up Top_Down • Two types of Language(5M) <ul style="list-style-type: none"> Procedure Oriented Object Oriented
5	<p>Explain Integration Test with example.(13M) (Nov/dec 2016)BTL3</p> <p>Answer:Page : 107 - Srinivasan & Ramaswamy</p>

	<p>Explanation:</p> <ul style="list-style-type: none"> • Goals(2M) • Integration Strategies: <ul style="list-style-type: none"> i. Top – Down ii. Bottom – Up iii. Bi – Directional • Designing Integration Test: <ul style="list-style-type: none"> 1. Black Box Approach • White Box Approach • Integration test strategy for procedures(5M) • Integration test strategy for classes(6M) • Critical Module characteristics • Example : Sandwich Testing
6	<p>Explain the different types of system testing with example.(13M)BTL2</p> <p>Answer:Page : 130 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Functional testing(1M) • Performance testing(1M) • Stress testing(1M) • Configuration testing(1M) • Security testing(1M) • Recovery testing(1M) • Fig: Types of System Test(4M) • Fig: Example of special resources needed for a performance test(3M)
7	<p>Explain in detail about scenario Testing.(13M)BTL2</p> <p>Answer:Page : 130 - Srinivasan & Ramaswamy</p> <ul style="list-style-type: none"> • Two Methods(4M) <ul style="list-style-type: none"> i. System Scenarios ii. Use Case Scenarios • Why Scenario test?(4M) <ul style="list-style-type: none"> i. Learn product ii. Connect Testing to documented requirement iii. Expose failure to deliver described benefits iv. Expose expert use of program v. Bring requirement related issues • Twelve ways to create good scenarios(5M)
7	<p>How would you identify hardware and software for configuration testing and how would you apply website testing?(13M)(Nov/dec 2016)BTL5</p> <p>Answer:Page : 195,198,369 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Configuration testing - testing application with multiple combinations (7M) • To find out the optimal configurations • Web testing - focuses on web applications.(6M) • Complete testing of web-based system before going live • Help address issues before system revealed to the public.

8	<p>i) Explain about Defect Bash Elimination.(7M)BTL2</p> <p>Answer : Page : 39 – Notes</p> <ul style="list-style-type: none"> • Ad-hoc Testing(2M) • Not based on written test cases(2M) • Brings together plenty of good practices(1M) • Steps in defect bash(2M) <p>ii) Explain about Ad-hoc Testing in detail.(6M)BTL2</p> <p>Answer : Page : 39 – Notes</p> <ul style="list-style-type: none"> • Discovers unfound errors in software(2M) • Impacted due to(2M) <ul style="list-style-type: none"> i. Intuition ii. Previous Experience iii. Expert knowledge of the platform iv. Experience in Testing • Drawback • Figure : Ad - hoc Testing(2M)
9	<p>i) Explain about usability and accessibility Testing.(7M)BTL2</p> <p>Answer : Page : 49 - Notes</p> <p>Usability testing:(4M)</p> <ul style="list-style-type: none"> • Characteristics • Quality Factors • Approach to usability • Aesthetic testing <p>Accessibility Testing:(3M)</p> <ul style="list-style-type: none"> • Basic accessibility • Product accessibility <p>ii) Explain Testing OO Model in detail.(6M)(BTL2)</p> <ul style="list-style-type: none"> • Unit Testing • Integration testing • Validate and system testing • Regression testing

10	<p>i) Differentiate Alpha and Beta Testing and discuss the phases in which alpha and beta testing are done?(7M)</p> <p>ii) Explain about documentation testing in detail.(6M)(Nov/Dec 2017)BTL3</p> <p>Answer:Page : 137-140 - Srinivasan & Ramaswamy</p> <p>Alpha Testing: (4M)</p> <ul style="list-style-type: none"> • Type of acceptance testing • Performed to identify all possible issues/bugs • Before releasing the product to everyday users or public. • Aim to carry out the tasks that a typical user might perform. <p>Beta Testing:(3M)</p> <ul style="list-style-type: none"> • Second phase of Software Testing • Sampling of the intended audience tries the product out. • Beta Testing of a product is performed by real users of the software application in a real environment. <p>ii)Explanation:</p> <ul style="list-style-type: none"> • Importance of documentation testing • Main things to look for in reviewing the document • Packaging and text graphics • Marketing materials,ads and other inserts • Warranty/Registration • EULA • Label and stickers • Installation setup & Instructions • Users Manual • Online help
----	--

PART – C

1	<p>If you are assigned to test compatibility of your product's data file formats, How would you approach the task?(15M)BTL6</p> <p>Answer:Appendix - Srinivasan Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Research whether your program follows existing standards for its files.(6M) • If so, test that it meets its standards.(1M) • Equivalence partition the possible programs that would read and write your program's files.(6M) • Design test documents with representative sample of the types of data.(2M)
2	<p>Explain the significance of control flow graph and cyclomatic complexity with the pseudo code for the sum of n numbers(13M).(Nov/Dec 2017)BTL6</p> <p>Answer : Appendix - Srinivasan Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Cyclomatic complexity is a software metric used to measure the complexity of a program.(5M)

	<ul style="list-style-type: none"> • This metric measures independent paths through the program's source code. An independent path is defined as a path that has at least one edge which has not been traversed before in any other paths.(6M) • Cyclomatic complexity can be calculated with respect to functions, modules, methods or classes within a program.(4M)
3	<p>What basic elements of a web page can easily be tested with black box Approach?(15M)BTL6</p> <p>Answer : Appendix - Srinivasan Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • The elements that are similar to what is multimedia CD – ROM software – text graphics ,and hyperlinks.(3M) • Web testing - focuses on web applications.(6M) • Complete testing of web-based system before going live (3M) • Help address issues before system revealed to the public.(3M)

JIT - 2106

UNIT IV TEST MANAGEMENT	
People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.	
PART* A	
1	Define Goal in testing. BTL1 A Goal can be described as a statement of intent or a statement of accomplishment of an individual test person.
2	What are the three types of goals in testing? BTL1 <ul style="list-style-type: none"> • Business Goal • Technical Goal • Political Goal
3	Define the term policy.(Nov/Dec2016) BTL1 A policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an organization.
4	Define Test Plan.(Nov/Dec 2015) BTL1 A Plan is a document that provides a frame work or approach for achieving a set of goals.
5	List the various Test Plan components.(Nov/Dec2016) BTL1 <ul style="list-style-type: none"> • Test Plan identifier • Introduction • Items to be tested • Features to be tested • Pass/Fail criteria • Suspension & Resumption criteria • Testing tasks Test environment • Risks & Contingencies • Testing costs • Approvals
6	Define Features. BTL1 Features may be described as distinguishing characteristics of a software component or system.
8	What is the meaning of the term Pass / Fail Criteria? BTL1 Given a test item and a test case, the tester must have a set of criteria to decide on whether the test has been passed or failed upon execution.
9	What is Suspension & Resumption criteria? BTL1 The criteria to suspend and resume testing are described in the simplest of cases testing is suspended at the end of a working day and resumed the following morning.
10	Define Work Breakdown Structure (WBS). BTL1 A Work Break Down structure is a hierarchical or tree like representation of all the tasks that are required to complete a project.
11	Define Risks & Contingencies. BTL1 Every testing effort has risks associated with it. Testing software with a high degree of critically, complexity, or a tight delivery deadline all impose risks that may have negative impacts on project goals.
12	What is Cost Drive? BTL1

	A Cost Driver can be described as a process or product factor that has an impact on overall project costs.
13	What are the various components of the test plan? AU Nov/Dec 2016 BTL1 <ul style="list-style-type: none"> • Test Design Specification • Test Case Specification • Test Procedures specifications
14	Define Test Summary Report. BTL1 This report is prepared when testing is complete. It is summary of the results of the testing efforts. It also becomes a part of the projects historical database and provides a basis for lessons learned as applied to future projects.
15	List the skills needed by a Test specialist. BTL1 <ul style="list-style-type: none"> • Organizational and planning skills • The ability to keep track of and pay attention to details • The determination to discover and solve problems • The ability to mentor and train others • The ability to work with users and clients • The ability to think creatively
16	What is the use of V-model in testing? BTL1 The V-model is model that illustrates how testing activities can be integrated in to each phase of the standard software life cycle.
17	Write the WBS elements for testing. BTL1 <ul style="list-style-type: none"> • Project start-up • Management coordination • Tool selection • Test planning • Test design • Test development • Test execution • Test measurement, and monitoring • Test analysis and reporting • 10. Test process improvement
18	What is the function of Test Item Transmittal Report or Locating Test Items? BTL2 Suppose a tester is ready to run tests on the data described in the test plan. We needs to be able to locate the item and have knowledge of its current status. This is the function of the Test Item Transmittal Report. Each Test Item Transmittal Report has a unique identifier.
19	Define Test Log. BTL1 The Test log should be prepared by the person executing the tests. It is a diary of the events that take place during the test. It supports the concept of a test as a repeatable experiment.
20	What are the Three critical groups in testing planning and test plan policy? (April/May 2015) BTL1 <ul style="list-style-type: none"> • Managers: • Developers/Testers • Users/Clients
21	What is scenario Testing? The process of giving the usage scenario of the system in the client's point of view and checking how the system reacts to it is called as scenario Testing.

22	What are the information present in the Test Item Transmittal Report or Locating Test Items?BTL1 <ul style="list-style-type: none"> • Version/revision number of the item • Location of the item • Person responsible for the item (the developer) • References to item documentation and test plan it is related to. • Status of the item • Approvals – space for signatures of staff who approve the transmittal.
23	What are the skills needed by a test specialist?BTL1 <ul style="list-style-type: none"> • Personal and managerial Skills <ul style="list-style-type: none"> • Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively. • Technical Skills <ul style="list-style-type: none"> • General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.
24	Write the test term hierarchy?BTL2 <ul style="list-style-type: none"> • Test Manager • Test leader • Test Engineer • Junior Test Engineer
25	Write the approaches to test cost Estimation?BTL2 <ul style="list-style-type: none"> • The COCOMO model and heuristics • Use of test cost drivers • Test tasks • Tester/developer ratios • Expert judgment
	PART* B
1	Explain the role of the 3 critical groups in software testing. (13M)BTL2 Answer:Page: 321 - Srinivasan & Ramaswamy <ol style="list-style-type: none"> 1. Managers(4M) <ul style="list-style-type: none"> • Task forces,policies,standards • Planning • Resource allocation • Support for education and training • Interact with users 2. Developers/ testers(5M) <ul style="list-style-type: none"> • Apply black and white box methods • Assist with test planning • Test at all levels • Train and mentor • Participate in task forces • Interact with users 3. Users/clients(4M) <ul style="list-style-type: none"> • Specify requirements clearly • Participate in usability test

2	<p>Explain the various documents involved in reporting Test Results. (13M)BTL2</p> <p>Answer: Page : 59 - Notes</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Test log(1M) • Test log identifier(2M) • Description(1M) • Activity and event entities(1M) • Test incident report(3M) • Test incident report identifier(1M) • Summary(1M) • Impact(1M) • Test summary report(2M)
3	<p>Explain the various Test Plan attachments? (13M)BTL2</p> <p>Answer: Page : 381 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Test design specifications(4M) • Test case specifications(5M) • Test procedure specifications(4M)
4	<p>Discuss in detail about the test plan components.(13M)(Nov/Dec 2016,Nov/Dec 2017)</p> <p>BTL2 Answer: Page : 59 – Notes</p> <p>Test Plan Components(13M)</p> <ul style="list-style-type: none"> • Test plan identifier(5M) • Introduction • Items to be tested • Features to be tested • Approach • Pass/fail criteria(4M) • Suspension and resumption criteria • Test deliverables • Testing tasks • Test environment • Responsibilities(4M) • Staffing and training needs • Scheduling • Risks and contingencies • Testing costs • Approvals
5	<p>Evaluate the testing and debugging goals and policies in detail.(13M)(April/May 2017)BTL5</p> <p>Answer: Page : 62 - Notes</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Debugging goal (4M) • Debugging policy(4M) • Testing Policy: Organization X(3M) • Debugging policy: Organization X(2M)
6	<p>Describe Test planning in detail. (13M) BTL2</p> <p>Answer: Page : 352 - Srinivasan & Ramaswamy</p>

	<p>Explanation:</p> <ul style="list-style-type: none"> • Planning(1M) • Milestone (1M) • Overall test objectives(2M) • What to test (Scope of the tests) (1M) • Who will test? (2M) • How to test? (2M) • When to test? (2M) • When to stop Testing? (2M)
7	<p>Explain in detail about Mutation testing. (13M) (April/May 2017) BTL2</p> <p>Answer: Page : 58 - Notes</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Mutation testing is a method of software testing in which program or source code is deliberately manipulated(4M) • Followed by suite of testing against the mutated code(5M) • The mutations introduced to source code are designed to imitate common programming errors.(4M)
8	<p>Discuss in detail about the various skills needed by test specialist.(13M) (Nov/dec2017)BTL2</p> <p>Answer:Page : 352 - Srinivasan & Ramaswamy</p> <ul style="list-style-type: none"> • Personal and managerial Skills(7M) <ul style="list-style-type: none"> ➢ Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively. • Technical Skills(6M) <ul style="list-style-type: none"> • General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.
9	<p>Explain the organizational structure for testing in single product companies.(13M) BTL2 (April/May 2017)</p> <p>Answer:Page :321 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Exploits the rear loading nature of testing activities.(2M) • Enables Engineers to gain experience in all aspects of life cycle(4M) • Is amenable to the fact that the organization mostly has informal processes.(2M) • Some defects may be detected earlier.(3M) • Accountability for testing quality reduces.(1M) • Schedule pressures normally compromise testing.(1M)
	PART* C
1	<p>Describe pesticide paradox and how bring in new people to look at the software helps solve it.(15M)BTL5</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <p>This is the situation that occurs if you continue to test (3M)</p> <ul style="list-style-type: none"> • Software with the same tests or same people.(4M) • Eventually, the software seems to build up immunity to the test because no new bugs are found.(3M)

	<ul style="list-style-type: none"> • If you change the tests or bring in new testers ,you will find new bugs.(2M) • The bugs are already there,it's the new technique which made the bugs visible.(3M)
2	<p>Why is the process of creating the test plan matters ,not the plan itself?(15M)BTL5</p> <p>Answer:Page: Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> • Because all the issues and the questions defined in the test plan either impact or influenced by other project functional groups or team members.(4M) • Getting everyone to understand and agree to the contents of the plan is what matters.(4M) • Privately creating a paper document and putting it on a shelf is not just a waste of time, but also jeopardizes the project.(7M)
3	<p>Justify the statement “ A schedule should be made to meet absolute dates ,so that there s no question when a testing task or phase is to start and when it is to end”.(15M)BTL6</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> • The statement is false (3M) • Because testing depends so much on other aspects of the project(5M) • For example ,you can't test something until its coded), a test schedule is best made relative to the delivery status.(7M)
4	<p>Name a few typical testing resources that should be considered when test planning.(15M)BTL6</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> • People, Equipment, Offices, Labs , Software ,Outsourcing Companies and miscellaneous supplies.(3M) • What are the entrance and exit criteria?(4M) • The requirements must be met to move from one testing place to another.(3M) • A Phase can't be left until its exit criteria are met.(3M) • A new phase can't be entered until its entrance criteria are met.(2M)

JIT

UNIT V TEST AUTOMATION	
Software test automation – skill needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.	
PART * A	
1	Define the term Project monitoring. BTL1 <ul style="list-style-type: none">• Project Monitoring refers to activities and tasks managers engage in to periodically check the status of each project.• Reports are prepared that compare the actual work done to the work that was planned.
2	Define the term Project controlling. BTL1 Project Controlling consists of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned.
3	Define Milestones. (Nov/Dec2016) BTL1 Milestones are tangible events that are expected to occur at a certain time in the project's lifetime. Managers use them to determine project status.
4	Differentiate version control and change control. BTL2 <ul style="list-style-type: none">• Version Control combines procedures and tools to manage different versions of configuration objects that are created during software process.• Change control is a set of procedures to evaluate the need of change and apply the changes requested by the user in a controlled manner.
5	What are the goals of Reviewers? BTL1 <ul style="list-style-type: none">• Identify problem components or components in the software artifact that need improvement.• Identify components of the software artifact that do not need improvement.• Identify specific errors or defects in the software artifact.• Ensure that the artifact conforms to organizational standards.
6	What are the benefits of a Review program? BTL1 <ul style="list-style-type: none">• Higher quality software• Increased productivity• Increased awareness of quality issues• Reduced maintenance costs• Higher customer satisfaction
7	What are the Various types of Reviews? BTL1 <ul style="list-style-type: none">• Inspections• Walk Throughs
8	Conclude on the need of Integration testing.(AU Nov/Dec2016) BTL2 <ul style="list-style-type: none">• Component integration testing that checks the interconnections between various parts (components) in a product.• System integration testing that tests the connections between the product and external systems.
9	What is Inspections? BTL1 It is a type of review that is formal in nature and requires prereview preparation on the

	part of the review team. The Inspection leader prepares is the checklist of items that serves as the agenda for the review.
10	What is Walkthrough?(Nov/Dec 2017) BTL1 It is a type of technical review where the producer of the reviewed material serves as the review leader and actually guides the progression of the review .It have traditionally been applied to design and code.
11	List out the members present in the Review Team.BTL1 <ul style="list-style-type: none"> • SQA(Software Quality Assurance) staff • Testers • Developers • Users /Clients. • Specialists.
12	List the components of review plans.(AU April/May 2015)BTL1 <ul style="list-style-type: none"> • Review Goals • Items being reviewed • Preconditions for the review. • Rolls, Team size, participants • Training requirements. • Review steps. • Time requirement
13	What are the advantages of review approach.BTL1 There are two pass approach for detect detection. <ul style="list-style-type: none"> • Pass 1 has individuals first reading reviewed item • Pass 2 has the item read by the group as a whole.
14	What are the various roles in review program?BTL1 <ul style="list-style-type: none"> • Review Leader • Review Recorder • Reader Reviewer
15	List the various review team membership constituencyReview Team Members.BTL1 <ul style="list-style-type: none"> • SQA Staff • Testers • Developers • Users / Clients • Specialists
16	What are the various different types of software artifacts.BTL1 <ul style="list-style-type: none"> • Requirement Reviews • Design Reviews • Code Reviews • Test Plan reviews
17	Define Change Control Board (CCB).BTL1 <ul style="list-style-type: none"> • There are 2 aspects of change control – one is tool based, the other term based. • The team involved is called CCB.
18	Define Project monitoring.BTL1 Project monitoring refers to the activities and tasks managers engage into periodically check the status of each project.Reports are prepared that compare the actual work done to the work that was planned or tracking.

19	<p>Define Project Controlling.BTL1 It is the process of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned.</p>
20	<p>Define Defect Removal Leverage (DRL).BTL1 This is a ratio of the defect detection rates from two review or test phases and can be expressed as</p> $DRL = \frac{\text{Defects / hour (review or test phase X)}}{\text{Defects / hour (review or test phase Y)}}$
21	<p>What are the various steps in the inspection process?BTL1</p> <ul style="list-style-type: none"> • Entry Criteria • Initiation • Preparation • Inspection Meeting • Reporting results • Rework & follow up
22	<p>What is the Role of process in Software quality?BTL1</p> <ul style="list-style-type: none"> • Capability Maturity Model. • Testing Maturity model (TMM)
23	<p>List the measurements and milestones for monitoring and controlling.BTL1</p> <ul style="list-style-type: none"> • Measurements for monitoring testing status • Coverage measures • Test case development • Test execution • Test harness development • Measurements to monitor tester productivity • Measurements for monitoring testing costs • Measurements for monitoring errors, faults, and failures • Monitoring test effectiveness
24	<p>Overview of the Testing Maturity Model(TMM)& the test related activities that should be done for V-model architecture.BTL1</p> <ul style="list-style-type: none"> • Test related issues • Benefits of test process improvement • Introduction to TMM • TMM levels
25	<p>List the criteria for test completion.BTL1</p> <ul style="list-style-type: none"> • All the planned tests that were developed have been executed and passed • All specified coverage goals have been met • The detection of a specific number of defects has been accomplished • The rates of defect detection for a certain time period have fallen below a specified

	level , Fault seeding ratios are favorable
--	--

PART * B

- | | |
|---|--|
| 1 | Illustrate with a sketch describe the design and architecture for test automation. (13M)(Nov/Dec ,2016)BTL1 |
|---|--|

Answer:Page : 396 - Srinivasan & Ramaswamy

Explanation:

- External modules.(3M)
- Scenario and configuration file modules.(3M)
- Test cases and test framework modules.(3M)
- Tools and results modules.(2M)
- Report generator and report metrics modules.(2M)

- | | |
|---|---|
| 2 | Explain the various generations of automations and the skills for each.(13M)(Nov/Dec,2017)BTL1 |
|---|---|

Answer:Page :392 - Srinivasan & Ramaswamy

Explanation:

- First Generation – Record and playback(4M)
- Second Generation – Data Driven(5M)
- Third Generation - Action Driven(4M)

- | | |
|---|--|
| 3 | Explain the design and architecture of test automation and list the challenges.(13M)(April /May ,2017).BTL2 |
|---|--|

Answer:Page :396 - Srinivasan & Ramaswamy

Explanation:

- External modules.(2M)
- Scenario and configuration file modules.(2M)
- Test cases and test framework modules.(2M)
- Tools and results modules.(2M)
- Report generator and report metrics modules.(2M)
- Challenges(3M)
- Certain types of testing cannot be executed without automation.
- Automation means end to end not test execution alone.

- | | |
|---|--|
| 4 | Discuss in detail about the controlling and monitoring: three critical views. (13M)BTL2 |
|---|--|

Answer:Page : 71 – Notes

Explanation:

- Measurements for monitoring testing status(1M)
- Coverage measures(1M)
- Test case development(2M)
- Test execution(1M)

	<ul style="list-style-type: none"> • Test harness development(2M) • Measurements to monitor tester productivity(2M) • Measurements for monitoring testing costs(1M) • Measurements for monitoring errors, faults, and failures(1M) • Monitoring test effectiveness(2M)
5	<p>Explain in detail about the role of reviews in testing software deliverables.(13M) BTL2</p> <p>Answer:Page : 68 - Notes</p> <p>Planning the Review (5M)</p> <ul style="list-style-type: none"> • The role and responsibilities of the review leader • Identifying the deliverable to review and its review criteria • Developing review checklists for the reviewers based on requirements • Selecting the review team and assign review duties <p>Conducting the Review (4M)</p> <ul style="list-style-type: none"> • The role and responsibilities of the review leader • Inform the reviewers of their review duties, tasks, and schedule • Collect the reviews in a review meeting • Dealing with interpersonal issues • Common review pitfalls and how to avoid them <p>Report and Follow-up on the Review(4M)</p> <ul style="list-style-type: none"> • The role and responsibilities of the review leader • Compile the review findings into a single review report • Track review findings or issues • Follow-up on review findings or issues
6	<p>Describe the various metrics and measurements in software testing and explain the various areas of metrics. (13M) (Nov/Dec 2016) BTL2</p> <p>Answer:Page: 420 - Srinivasan & Ramaswamy</p> <p>Explanation:</p> <ul style="list-style-type: none"> • Project metrics(2M) • Effort variance(3M) • Schedule Variance(3M) • Effort Distribution across phase(5M)
1	<p>How will you differentiate tools and automation? Name the few benefits and drawbacks of using software test tools and automation. (15M)BTL6</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> • A testing tool will help you test .making it easier for you to perform a manual testing task.(3M) • Automation is also a tool but it will run without your intervention.(3M) • Think power saw and hammer building a house while the carpenter sleeps.(3M) <p>Benefits:(3M)</p> <ul style="list-style-type: none"> • Speed up the amount of time it takes to run your test process. • Precise and relentless. <p>Drawbacks:(3M)</p>

	<ul style="list-style-type: none"> Because software can change during the product's development, your test tools will be need to change. It is easy to rely on automation much.
2	<p>If you were using metrics from the bug – tracking database to measure your progress or success at testing, why would just counting the number of bugs you find per day or computing your average find rate be an insufficient measure? (15M)BTL6</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> It does not tell the entire story. You could be testing the complex area of the software.(4M) Your area could have been written by the most experienced programmer.(4M) It could have been written by the least experienced programmer.(4M) The code that you are testing may already have been tested or may be brand new.(3M)
3	<p>“The test team is responsible for the quality of the product” Does the statement make sense, Justify your answer with necessary explanation. (15M)BTL6</p> <p>Answer:Page : Appendix - Srinivasan Ramaswamy</p> <ul style="list-style-type: none"> False! Testing looks for bugs .(7M) Testers didn't put the bugs in the product and can't guarantee when they are done testing that no more bugs exist.(8M)

JIT