CMR Institute of Tecl			
Department(s) : MCA			
Semester: 02 Section(s):			
Subject: Python Progr	amming	16MCA21	Lectures /Week :05
Course Instructor(s) :	Kavitha K		
Course Duration : Feb	oraury 2017 to May 2017		

Course Objectives: This subject provides students with

- 1. Basics python programming
- 2. Using Modules and functions
- 3. Using collections

Course Prerequisites:No prior knowledge of programming is required

Course Outcome (CO):

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

CO1: Understand and comprehend the basics of python programming.

CO2: Apply knowledge in real time applications.

CO3: Understands about files and its applications.

Lecture	Book &	Torrige	Portions	Coverage %
#	Sections	Topics	Individual	Cumulative
1 to 10	TB1: 1-73	Module1 : Introduction:InstallingPython, Simple program using Python,Expressions and Values, Variables andComputer Memory, error detection,Multiple line statements, Designing andusing functions, functions provided byPython, Tracing function calls in memorymodel, omitting return statement.Working with Text: Creating Strings ofCharacters, Using Special Characters inStrings, Creating a Multiline String,Printing Information, Getting Informationfrom the Keyboard.	10	20
11 to 20	TB1: 77-123	Module 2:A Boolean Type , Choosing Statements to Execute, Nested If Statements , Remembering the Results of a Boolean Expression Evaluation , A Modular Approach to Program Organization, Importing Modules ,	20	40

		Defining Your Own Modules, Testing Code Semi automatically Grouping Functions Using Methods: Modules, Classes, and Methods, Calling Methods the Object- Oriented Way, Exploring String Methods, Underscores.		
21 to 30	TB1: 129-161	Module 3: Storing Collections of Data Using Lists: Storing and Accessing Data in Lists, modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops Using Break and Continue Reading and Writing	10	60
31 to 40	TB1: 171-218	Module 4:Kinds of files, Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, and Writing Algorithms That Use the File-Reading Techniques, Multiline Records. Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, Inverting a Dictionary, Using the In Operator on Tuples, Sets, and Dictionaries, Comparing Collections.	10	80
41 to 50	TB1: 269-293 317-336	Module 5:Collection of New Information Object-Oriented Programming : Understanding a Problem Domain , Function "Isinstance," Class Object, and Class Book , Writing a Method in Class Book, Plugging into Python Syntax: More Special Methods ,Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style Introducing few more Widgets, Object-Oriented GUIs, Keeping the Concepts from Being a GUI Mess.	10	100

Syllabus for Internal Assessment Test

Internal Assessment Test	Syllabus
T1	Class # 01 - 24

T2 T3		Class # 25 - 48 Class # 49 - 62		
- •				
Literature Code		Author & Tit	Publication In	ıfo
Book Type	Coue	Autiloi & Itt	Edition & Publisher	ISBN #
Text Book	TB1	Paul Gries , Practica Programming	Edition Pragmatic	SBN-13: 978- -93778-545-1
Text Book	TB2	Allen Downey,Lear Python-How to thin a Computer Scienti	k like Green Tea Publisher	ISBN 0- 9716775-0-6

	CMR Institute of Technology, Bangalore Department(s):Master of Computer Applications			
Semester: 02	Section : A		CMR INSTITUTE OF TECHNOLOGY	
Object Oriented Programmir	ng using C++	16MCA22	Lectures/week: 06	
Course Instructor(s): Ms. Ne	ha Agrawal			
Course duration : 6 Feb 2017	7 – 21 May 2017			

Course Objectives:

The course presents basics of C++ programming including: Basics of C++ environment, Data representation, Control structures, Functions, Arrays, Pointers, Strings, and Classes that aims to:

- 1. Arm the students with the basic programming concepts.
- 2. Introduce different techniques pertaining problem solving skills
- 3. Arm the students with the necessary constructs of C++ programming.
- 4. And to emphasis on guided practical sessions

Prerequisites:

C programming language. Concepts of variables, data types, functions, pointers

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Differentiate between object oriented programming and procedure oriented programming & Disseminate the importance of Object oriented programming

CO2: Apply C++ features such as Classes, objects, constructors, destructors, inheritance, operator overloading, and Polymorphism, Template and exception handling in program design and implementation.

CO3: Use C++ to demonstrate practical experience in developing object-oriented solutions. **CO4:** Analyze a problem description and build object-oriented software using good coding practices and techniques.

CO5: Implement an achievable practical application and analyze issues related to object-oriented techniques in the C++ programming language

Class #	Chapter Title / Reference	Торіс	Percentage cove	-
	Literature		Reference	Cumulativ e
1		Overview of C++: The Origins of C++, What Is Object-Oriented Programming? ,Encapsulation ,Polymorphism , Inheritance. Some C++ Fundamentals		
2	TB1 : 11	A Sample C++ Program ,A Closer Look at the I/O Operators, Declaring Local Variables ,No Default to int, The bool Data Type		
3	Chapter	Old-Style vs. Modern C++ , The New C++ Headers , Namespaces, Working with an Old Compiler	9%	9%
4	_	Introducing C++ Classes ,Function Overloading, Operator Overloading		
5	-	Inheritance ,Constructors and Destructors , The C++ Keywords, The General Form of a C++ Program.		
7 8	-	Classes and Objects: Classes, Structures and Classes Are Related, Unions and Classes Are Related, Anonymous Unions Friend Functions, Friend Classes, Inline Functions, Defining Inline Functions Within a Class		
9	TB1 :12	Parameterized Constructors, Constructors with One Parameter: A Special Case Static Class Members ,Static Data Members ,Static	11%	21%
10 11	chapter	State Class Members , state Data Members , state Member Functions When Constructors and Destructors Are Executed The Scope Resolution Operator ,Nested Classes, Local		
12 13	-	Classes, Passing Objects to Functions ,Returning Objects ,Object Assignment.		
14 15	_	Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Creating Initialized vs. Uninitialized Arrays, Pointers to Objects, Type Checking C++ Pointers, The this Pointer, Pointers to Derived Types		
16	TB1 :13	Pointers to Class Members , References, Reference Parameters, Passing References to Objects	9%	200/
17	chapter	Returning References, Independent References, References to Derived Types, Restrictions to References	9%	30%
18		A Matter of Style, C++'s Dynamic Allocation Operators ,Initializing Allocated Memory ,Allocating Arrays		
19		Allocating Objects ,The nothrow Alternative ,The Placement Form of new		
20 21	-	Function Overloading, Copy Constructors and Default Arguments: Function Overloading Overloading Constructors		
22	TB1 :14 -chapter	Overloading a Constructor to Gain Flexibility, Allowing Both Initialized and Uninitialized Objects Copy Constructors	9%	40%
23 24		Finding the Address of an Overloaded Function , The overload Anachronism Default Function Arguments, Default Arguments vs. Overloading		
25		Function Overloading and Ambiguity.		
26	TB1 :15 Chapter	Operator Overloading: Creating a Member Operator Function	9%	50%

27 and decrement Operators 28 Overloading the Shorthand Operators, Operator, Operator Overloading Using a Friend Punction, Using a Printed to Overloading the shorthand Operator, Operator Overloading the Shorthand Operator, Operator Overloading the onthrow Version of new and delete. 29 Overloading Using a Friend Punction, Using a Printed to Overloading the onthrow Version of new and delete. 30 Overloading Conse Special Operators, Overloading [1]. 31 Overloading (1), Overloading =>, Overloading the Overloading (1), Overloading =>, Overloading 11. 32 Inheritances: Base-Class Access Control, Inheritance and Overleted Meeters 33 TB1:16 Inheritances: Base-Class Access Control, Inheritance (1), Overloading pranettors to Base-Class Constructors and Destructors Are Executed 9% 60% 34 TB1:16 Inheritance Ibase-Class Constructors 9% 60% 35 Chapter Constructors and Destructors Are Executed 9% 60% 36 TB1:17 Pure Virtual Aurinbute Is Inherited, Virtual Functions Are Flearachical 9% 60% 39 TB1:17 Pure Virtual Aurinbute Is Inherited, Virtual Function with Two Generic Punctions, Claing a Paray Polying Ceneric Punctions, Claing a Paray Polying Ceneric Punctions, Claing a Paray Polying Template Classes: A Generic Punctions, Restricting August Phyling Ceneric Punctions, Cathing all Exceptions, Restricting August Ph	1		Creating Prefix and Postfix Forms, of the Increment		
28 Overloading Restrictions 29 Operator Overloading Using a Friend Function, Using a Friend to Overload ++ or, Friend Operator Functions Add Flexibility 30 Overloading new and delete. Overloading new and delete for Arrays, Overloading ->, Overloading the Overloading Some Special Operators, Overloading []. 31 Overloading Some Special Operators, Overloading the Overloading (), Overloading ->, Overloading the Comma Operator. 32 Inheritances: Base-Class Access Control, Inheritance and protected Members 33 Protected Base-Class Constructors Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed 9% 36 Prostected Base-Class Constructors 9% 37 Granting Access, Virtual Base Classes. 9% 38 Virtual Functions, Alling a Virtual Function Restructors, Calling a Virtual Functions Are Hierarchical 5% 40 chapter The Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 41 Functions, Alling a Virtual Function Template Using Standard Parametors with Template Functions, Generic Function, Applying Generic Function, Dverloading a Function Template Using Standard Parametors with Template Functions, Generic Function, Acchange a Function Functionate Classes, Using Non-Type Arguments with Template Classes, Generic Function, Scatching a Start Types, Using Multiple catch Statenents 6%	27		-		
29 Operator Overloading Using a Friend Function, Using a Friend to Overloading new and delete for Arrays, Overloading new and delete for Arrays, Overloading the nothrow Version of new and delete Overloading Some Special Operators, Overloading [1], Overloading (1), Overloading ->, Overloading the Comma Operator. 31 Overloading Case Special Operators, Overloading [1], Overloading (1), Overloading ->, Overloading [1], Overloading (1), Overloading ->, Overloading the Comma Operator. 9% 33 Protected Base-Class Control, Inheritance and protected Base-Class Control, Inheritance and protected Base-Class Constructors. 9% 36 Protected Base-Class Constructors. 9% 37 Granting Access, Virtual Base Classes. 9% 38 Virtual Functions, and Delymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 9% 39 TB1:17 Pure Virtual Functions, Abstract Classes, Using Virtual Function, Overloading a Generic Function, State Protections, A Function with Two Generic Types, Explicitly Overloading a Generic Function, A Ceneric Sort, Compacting an Array Classes: Generic Functions, Applying Generic Functions, A Ceneric Sort, Compacting an Array Classes, The Purgenters with Two Generic Data Classes, Applying Template Classes, A Classes, Publite Classes, The Publit, Class Specializations, The typename and export Keywords, The Public Togenset: A Generic Array Classes, Sing Non-Type Arguments with Generic Functions, A Clesnic Sort, Compacting an Array Classes, The Overloading a Exception, Handling Derived-Class Exceptions, Retricting Exception Flandling Fundamentals,			Overloading the Shorthand Operators, Operator		
29 a Friand to Overload#+ or 7 -, Friend Operator 30 Functions Ad Flexibility 30 Overloading new and delete, Overloading new and delete, Overloading Some Special Operators, Overloading IT, Overloading (), Overloading Some Special Operators, Overloading IT, Overloading C), Overloading Some Special Operators, Overloading IT, Overloading (), Overloading Some Special Operators, Overloading IT, Overloading C), Overloading Some Special Operators, Overloading IT, Overloading C, Overloading S, Overloading IT, Overloading C, Overloading S, Overloading IT, Overloading C, Overl	28				
29 Functions Add Flexibility 30 Overloading new and delete, Overloading new and delete for Arrays, Overloading the northrow Version of new and delete for Arrays, Overloading the northrow Version of new and delete for Arrays, Overloading 1, O			Operator Overloading Using a Friend Function, Using		
30 Overloading new and delete. Overloading new and delete for Arrays, Overloading the nothrow Version of new and delete. 31 Overloading Some Special Operators. Overloading 1. Overloading C), Overloading ->, Overloading the Comma Operator. 32 Inheritances: Base-Class Access Control, Inheritance and protected Members 33 TB1:16 34 TB1:16 35 Protected Base-Class Inheritance and protected Members 36 Onstructors, and Destructors, and Inheritance, When Constructors and Destructors and Inheritance, When Constructors and Destructors and Inheritance, When Constructors and Destructors, and Inheritance, When Constructors and Delymorphism: Virtual Functions, Caling a Virtual Function Through a Base, Class Reference 39 TB1:17 Pure Virtual Functions, Abstract Classes, Using Virtual Functions, Caling a Virtual Function with Two Generic Types, Explicit() Overloading a Generic Function, Aparameters with Template Functions, Generic Types, Explicit() Overloading Generic Function, Applying Generic Function, Applying Template Classes: A Generic Array Class. Using Non-Type Arguments with Generic Functions, Applying Generic Functions, Applying Template Classes, A destawa Classes: A Destructor Array Class. Using Non-Type Arguments with Generic Functions, Catenic All Exception Adults of Statemard IL Exceptions, Retricting Exception Handling Fundamentals, Catching Classes: A Generic Array Class. Using Non-Type Arguments with Generic Function Restrictions, Applying Generic Function Restrictions, Applying Generic Function Restrictions, Applying Generic Function, Scatching All Exceptions, Restricting Exception Handling F	29				
30 new and delete' 31 Overloading Some Special Operators, Overloading [], Overloading (), Overloading ->, Overloading the Comma Operator. 32 Inheritances: Base-Class Access Control, Inheritance and protected Members 33 TB1:16 34 TB1:16 35 Protected Base-Class Inheritance 36 Inheriting Multiple Base Classes 36 Constructors, and Destructors, and Inheritance, When Constructors and Destructors and Secure Secure Class Constructors 37 Granting Access, Virtual Base Classes. 38 Virtual Functions and Polymorphism: Virtual Functions, Caling a Virtual Base Classes, Using Virtual Functions, Caling a Virtual Functions the Virtual Functions are class Reference 39 TB1:17 Pure Virtual Functions, A Explicitly Overloading Generic Function, New Scale Binding. 41 Generic Types, Explicitly Overloading Generic Function, New Fourie Min Template Functions are Generic Types, Applying Template Classes; A Generic Array Class. Using Non-Type Arguments with Generic Functions, A Generic Sort, Compacting an Array Class Using Non-Type Arguments with Generic Classes 6% 42 TB1:18 Generic Sort, Compacting an Array Class Using Non-Type Arguments with Generic Classes 6% 44 Exception Handling Functions, Catching AI LExceptions, Restricting Exception Handling Functions, Catching AI LExceptions, Restricting Exception Rest			Overloading new and delete, Overloading new and		
31 Overloading Some Special Operators. Overloading [1]. 31 Overloading (-), Overloading ->, Overloading the Comma Operator. 32 Inheritance: Base-Class Access Control, Inheritance: and protected Members 33 Protected Base-Class Access Control, Inheritance: and protected Members 34 TB1:16 Inheriting Multiple Base Classes 36 Constructors, Destructors, and Inheritance, When Constructors and Destructors. And Inheritance, When Constructors and Destructors and Inheritance, When Constructors and Polymorphism: Virtual Functions, Calling a Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function With Two Generic Types, Explicitly Overloading a Generic Function, Attribute Is Inherited, Virtual Functions, Are Hierarchical 5% 40 chapter Functions, Early vs. Late Binding. 5% 41 Templates: Generic Encloses, A Intexton with Two Generic Types, Explicitly Overloading a Generic Functions, A Generic Sort, Compacting an Array 6% 72% 42 TB1:18 Generic Sort, Compacting an Array 6% 72% 43 TB1:19 Retrievolve, Classes, An Example with Two Generic Array Classes 6% 8% 81% 44 Exception Handling Functions, Classes, An Example with Template Classes, Explicit Classes 6% 8% 81%	20		delete for Arrays, Overloading the nothrow Version of		
Overloading (), Overloading ->, Overloading the ⁻ 2 Inheritance: Base-Class Access Control, Inheritance and protected Members 33 Protected Base-Class Access Control, Inheritance 34 TB1:16 Inheritance: Base-Class Inheritance 34 TB1:16 Constructors, Destructors, and Inheritance, When Constructors, Sortauctors, and Inheritance, When Constructors, Sortauctors, and Inheritance, When Constructors, Sortauctors, and Inheritance, When Constructors, Sortauctors, and Classes. 36 Granting Access, Virtual Base Classes. 37 Granting Access, Virtual Base Classes. 38 Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Functions Mase, Class Reference 39 TB1:17 Pure Virtual Attribute Is Inheritad, Virtual Functions, Are Hierarchical 5% 40 chapter Functions, Rarly vs. Late Binding. 5% 41 Functions, A Generic Functions, Applying Generic Function, A Parameters with Template Functions, Generic Tayes, Applying Generic Function, A Generic Classes, A Example with Two Generic Data Types, Applying Template Classes, Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 41 Functions, A Caching Class Types, Using Multiple catch Statements Exception Handling 8% 81% <	30	4			
32 Inheritances: Base-Class Access Control, Inheritance and protected Base-Class Inheritance 33 TB1:16 Inheriting Multiple Base Classes 34 TB1:16 Inheriting Multiple Base Classes 35 Constructors and Destructors: and Inheritance, When Constructors and Destructors and Base Classes. 9% 36 Passing Parameters to Base-Class Constructors 9% 37 Granting Access. Virtual Base Classes. 9% 38 Virtual Functions and Polymorphism: Virtual Functions, Caling a Virtual Function Through a Base, Class Reference 5% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions and Polymorphism: Virtual Functions, Early vs. Late Binding. 5% 40 chapter Function, Nethoding a Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Functions, A Generic Sort, Compacting an Array Generic Function, Nor-Type Arguments with Generic Data Types, Applying Template Classes: A Generic Array Classes, Data Types, Applying Template Classes: A Generic Array Classes, Explicit Class Specializations, The typename and export Mey Model Specific Array Classes, Explicit Class Specializations, The typename and export Mey Model. 6% 41 Functional Exception Handling Classes: A Generic Array Classes. Explicit Class Specializations, The typename and export of Templates Classes. Explicit Class Specializations, The typename and export ferminate and Lexporetor Handling. 6% <t< td=""><td></td><td></td><td>Overloading (), Overloading ->, Overloading the</td><td></td><td></td></t<>			Overloading (), Overloading ->, Overloading the		
32 and protected Members 33 TB1:16 Inheriting Multiple Base Classes chapter Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed 9% 35 Protected Base-Class Inheritance, When Constructors, and Destructors Are Executed 9% 36 Passing Parameters to Base-Class Constructors 9% 37 Granting Access, Virtual Base Classes, 9% 38 Virtual Functions, and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 39 TB1:17 Pure Virtual Attribute Is Inheritad, Virtual Functions Are Hierarchical 5% 40 chapter Functions, Early vs. Late Binding. 5% 41 Templates: Generic Functions, A Function with Two Generic Punctions, A Generic Cases, An Example with Template Functions, Generic Punctions, A Generic Cases, A Deplying Generic Functions, A Generic Cases, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Exception Handling: Exceptions, Reception, Setting the Terminate and Unexpected Handlers, The Approx K_Subork, The Power of Templates 8% 81% 44 Exception Handling 81% 81% 81% 45 Fundamentals, C	31		Comma Operator.		
33 TB1:16 Protected Base-Class Inheritance 9% 60% 34 TB1:16 Inheriting Multiple Base Classes 9% 60% 35 Constructors, and Destructors, and Inheritance, When Constructors and Destructors Are Executed 9% 60% 36 Passing Parameters to Base-Class Constructors 9% 60% 37 Granting Access, Virtual Base Classes. 9% 60% 38 Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 66% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions, Are Hierarchical 5% 66% 40 chapter Teme Virtual Attribute Vortoading a Generic Functions, Compacting an Array 5% 66% 41 Generic Types, Explicitly Overloading a Generic Functions, Reerier Sort, Compacting an Array Class. Using Non-Type Arguments with Two Generic Data types, Applying Template Classes: A Generic Array Classes 6% 72% 43 TB1:18 Generic Classes: An Example with Wordeneric Classes 6% 72% 44 Expertion Handling: Exception Handling 6% 72% 45 TB1:19 Rethrowing an Exception Handling 8% 81% 46	32				
34 TB1:16 Inheriting Multiple Base Classes 9% 60% 35 Constructors, Destructors, And Inheritance, When Constructors And Executed SA net Executed 9% 60% 36 Passing Parameters to Base-Class Constructors 9% 60% 37 Granting Access, Virtual Base Classes. 9% 60% 38 Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 66% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 66% 39 TB1:17 Pure Virtual Functions, Abstract Classes, Using Virtual Functions, Active Verloading a Generic Function, Overloading a Function Template Using Standard Parameters with Template Functions , Generic Function Restrictions , Applying Generic Functions, A Generic Canses; A Generic Array Class Using Non-Type Arguments with Generic Classes 6% 72% 41 Generic Classes; An Example with Two Generic Data Types, Applying Template Classes; A Generic Array Classes 6% 72% 43 Generic Classes; Catching Class Types, Using Multiple catch Statements 6% 8% 81% 44 Exception Handling: Craster Exception Infandling 8% 81% 45 Fundamentals, Catching Class Types, Using Multiple					
135 Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed 9% 60% 35 Passing Parameters to Base-Class Constructors 9% 60% 37 Granting Access, Virtual Base Classes. 9% 60% 38 Virtual Functions, and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 66% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 66% 40 chapter Function, Seneric Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function Restrictions Applying Generic Function Restrictions Applying Generic Function Restrictions Applying Generic Function Restrictions Applying Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Data Types, Applying Template Classes, A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 41 Exception Handling: Exception Handling Fundamentals, Catching All Exceptions, Restricting Exception S, The Pypename and export Keywords, The Power of Templates 6% 81% 43 Exception Handling: Exception Randling terminate() and unexpected() 8% 81% 44 Exception Handling: Exception Randling terminate() 8% 81% 45 Handling Derived-Class Exception, Backers, Open		TD1 .16			
35 Constructors and Destructors Are Executed 36 Passing Parameters to Base-Class Constructors 37 Granting Access, Virtual Base Classes. 38 Virtual Functions, and Polymorphism: Virtual Functions, Calling a Virtual Attribute Is Inherited, Virtual Functions 39 TB1 :17 Pure Virtual Attribute Is Inherited, Virtual Functions 40 chapter Functions, Early vs. Late Binding. 41 Templates: Generic Functions, A Function with Two Generic Function, Compacting a Generic Functions, Compacting an Array 6% 41 Generic Types, Explicitly Overloading a Generic Ciass, Using Standard Parameters with Template Functions, Generic Calsses, An Example with Two Generic Data Types, Applying Template Classes: A Generic Cata Types, Applying Template Classes: A Generic Cata Class, Using Non-Type Arguments with Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Cata Class, Using Mon-Type Arguments with Generic Cata Class, Using Non-Type Arguments with Generic Cata Supprovers, The Power of Templates 6% 72% 43 Exception Handling: Exception Handling Fundamentals, Catching All Exceptions, Restricting Exception, Restricting Exception, Mindule Exception, Restricting Exception, Restricting Class Exception, Restricting Exception And Restricting Exception, Restricting Ex				9%	60%
36 Passing Parameters to Base-Class Constructors 37 Granting Access, Virtual Base Classes. 38 Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 40 chapter Functions, Abstract Classes, Using Virtual Function, Sarphy vs. Late Binding. 5% 41 Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Actendaria Parameters with Template Functions, Generic Curcion Restrictions , Applying Generic Function, SA Generic Sort, Compacting an Array 6% 72% 42 TB1:18 Generic Classes, An Example with Two Generic Cases, Class Using Non-Type Arguments with Generic Class. Using Default Arguments with Template Classes, Exsplicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 43 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple class. Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 8% 81% 44 Exception Handling: Exception Handling Fundamentals, Catching Classes Sception Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 48 Generic Handling 8% 81%	35	enapter			
38 Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 66% 40 chapter Functions, Early vs. Late Binding. 5% 66% 41 Templates: Generic Functions, Abstract Classes, Using Virtual Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template Generic Tunction Restrictions , Applying Generic Function, Overloading a Function Template Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 81% 44 Exception Handling Fundamentals, Catching Class Exception Handling Derived-Class Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exception, Understanding terminate() and unexpected() 8% 81% 48 Standard C++ I/O Classes: Old vs. Modern C++ I/O, C++ Streams, For C++ Stream Classes, Applying Exception Handling 6% 87% 50 The exception and bad_exception Classes Applying Exception Handling 6% 87% 51 Fredefined Streams, Formattel I/O Formatting Using the ios Members, Setting the Format Flags	36	-	Passing Parameters to Base-Class Constructors		
38 Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference 5% 39 TB1:17 Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical 5% 40 chapter Functions, Early vs. Late Binding. 5% 41 Templates: Generic Functions, A bastract Classes, Using Virtual Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template Generic Types, Explicitly Overloading a Generic Function, Starses, An Example with Tromplate Functions, Generic Function Restrictions, Applying Generic Function, A Generic Star, Compacting an Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 TB1:18 Generic Classes, An Example with Template Classes, Exepticit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 8% 81% 45 Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 48 Standard C++ I/O Classes: Old vs. Modern C++ I/O, C++ Streams, For C++ Stream Classes, C++'s Predefined Streams, Formattel I/O 6% 87% 50 Freederined Streams, Formattel I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading < and >> 6% 87% 5	-	1			
Functions, Calling a Virtual Function Through a Base, Class Reference39TB1:17Functions, Calling a Virtual Functions5%39TB1:17Pure Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical5%40chapterFunctions, Early vs. Late Binding, Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Starger vs. Late Binding, Generic Function Restrictions, A plying Generic Function, Restrictions, A Generic Stat, Compacting an Array Class, Using Non-Type Arguments with Generic Data Types, Applying Template Classes: A Generic Data Types, Applying Template Classes: A Generic Classes, Explicit Classes, Changer Classes, Classes, Changer Classes, C	38				
39 The Virtual Attribute Is Inherited, Virtual Functions 5% 66% 39 TB1:17 Pure Virtual Functions, Abstract Classes, Using Virtual 5% 66% 40 chapter Functions, Early vs. Late Binding. 5% 66% 41 Templates: Generic Functions, A Function with Two Generic repres, Explicitly Overloading a Generic Function, Overloading a Function Template 6% 72% 42 TB1:18 Generic Classes, An Example with Template Functions and Types, Applying Template Classes: A Generic Chata Types, Applying Template Classes: A Generic Classe, Using Non-Type Arguments with Generic Classes. 6% 72% 43 Classes, An Example with Tromplate Classes, Lass Using Default Arguments with Generic Classes. 6% 72% 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements. 6% 72% 45 Fundamentals, Catching Class Types, Using Multiple catch Statements. 8% 81% 46 TB1:19 Rethrowing an Exception Receptions, Exception Handling Exception Restricting Exception Restricting Exception Restricting Classes Applying Exception Restricting Exception Restricting Classes Applying Exception Classes Applying Exception Restricting Exception Restricting Classes Applying Exception Restricting Classes Applying Exception Restrictions, Restricting Classes Classes Applying Exception Restricting Classes, Classes Applyin					
39 Are Hierarchical Pure Virtual Functions, Abstract Classes, Using Virtual 40 chapter Functions, Early vs. Late Binding. 41 Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template 6% 42 TB1:18 Generic Classes, An Example with Template Functions, A Generic Sort, Compacting an Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 export Keywords, The Power of Templates 8% 81% 46 TB1:19 Rethrowing an Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 8% 81% 46 TB1:19 Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 48 Usangaba Exception (Classes Applying Exception and bad_exception Classes Applying Exception Handling 8% 81% 49 Exception Bis formatted I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Formatted I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Formattel I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functi					
TB1:17 Pure Virtual Functions, Abstract Classes, Using Virtual 40 chapter Functions, Early vs. Late Binding. 41 Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template 42 Using Standard Parameters with Template Functions, Generic Casses, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Explicit Class Specializations, The typename and export Keywords, The Power of Template 6% 72% 44 Exception Handling: Exception Handling Fundamentals, Catching All Exception Handling Options, Catching All Exception Handling Derived-Class Exception, Exception Handling Derived-Class Exception, Exception Handling Options, Catching All Exceptions, Restricting Exception (Understanding terminate() and unexpected() 8% 81% 48 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, Fiormattel I/O 8% 81% 50 Predefined Streams, Formattel I/O 6% 87% 51 Formatting Using the ios Members, Setting the Formatting Suing Your Own Extractors, Creating Your Own Manipulator Functions 6% 87% 52 TB1:20 TB1:20 File, Reading and Writing Text Files, 6% 87%				5%	66%
40 chapter Functions, Early vs. Late Binding. 41 Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template Using Standard Parameters with Template Functions, Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes: 6% 72% 43 TB1:18 chapter Exception Handling: Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling: Exception Handling: Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions, Exception, Handling Options, Catching All Exceptions, Restricting Exception () Function 8% 81% 45 TB1:19 47 Rethrowing an Exception () Function The exception and bad_exception Classes Applying Exception Handling 8% 81% 48 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% 50 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% 51 TB1:20 The Exception shores, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% <td>39</td> <td>TD1 17</td> <td></td> <td></td> <td></td>	39	TD1 17			
41 Templates: Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template Using Standard Parameters with Template Functions, Generic Function Restrictions, Applying Generic Functions, A Generic Sort, Compacting an Array Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Generic Classes, An Example with Template Classes: Classes 6% 72% 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 6% 8% 45 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions 8% 81% 46 TB1:19 Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 47 Setting the Terminate and Unexpected Handlers, The uncaught_exception () Function The exception Handling 8% 81% 49 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 6% 87% 50 Fredefined Streams, Creating Your Own Extractors, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Inserters, Creating Your Own 6% 87% 51 Flags, Clearing Format Flags, Overloading << and >> C++ File I/O: fitrear> and the File Cla	40		-		
41 Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template 42 Using Standard Parameters with Template Functions, Generic Functions, A Generic Sort, Compacting an Array 6% 42 TB1:18 Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling: Exception Handling: Exception Handling Fundamentals, Catching All Exception, Handling Options, Catching Classes Applying Exception Handling 8% 81% 48 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 6% 87% 50 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own 6% 87% 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own 6% 87% 52 TB1 :20 and Closing a File, Reading and Writing Text Files,	40	cnapter			
41 Function, Overloading a Function Template Functions, Generic Function Restrictions, Applying Generic Functions, A Generic Sort, Compacting an Array 6% 42 TB1:18 Generic Function Restrictions, Applying Generic Functions, A Generic Casses, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes. 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 8% 81% 45 Restricting Exceptions, Exceptions, Restricting Exceptions, Catching All Exceptions, Restricting Exception, Inderstanding terminate() and unexpected() 8% 81% 48 Setting the Terminate and Unexpected Handlers, The uncaught_exception () Function 8% 81% 49 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 6% 87% 50 Formating Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% 51 Flags, Clearing Format Flags, Overloading << and >> 6% 87% 52 Treating Your Own Inserters, Creating Your Own			Generic Types, Explicitly Overloading a Generic		
42 Generic Function Restrictions, A Applying Generic Functions, A Generic Sort, Compacting an Array Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 8% 81% 45 Exception Generic Classes, Restricting Exceptions, Exception, Handling Options, Catching All Exceptions, Restricting Exceptions, Restricting Exception, Understanding terminate() and unexpected() 8% 81% 48 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formattel I/O 6% 87% 50 Frage, Clearing Format Flags, Overloading <	41	_	Function, Overloading a Function Template		
42 Functions, A Generic Sort, Compacting an Array 6% 72% 43 Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 6% 72% 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 6% 72% 44 Exception Handling: Fundamentals, Catching Class Types, Using Multiple catch Statements 8% 81% 45 Handling Derived-Class Exceptions, Restricting Exception, Understanding terminate() and unexpected() 8% 81% 48 Setting the Terminate and Unexpected Handlers, The uncaught_exception () Function 8% 81% 49 Exception Handling Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s 6% 87% 50 Fredefined Streams, Formatted I/O Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% 51 Flags, Clearing Format Flags, Overloading << and >> 6% 87% 52 Transpreaming Your Own Inserters, Creating Your Own 6% 87% 52 Transpreaming Your Own Inserters, Creating Your Own 6% 87%			Using Standard Parameters with Template Functions, Generic Function Restrictions Applying Generic		
chapter Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 45 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions 46 TB1 :19 47 Rethrowing an Exception, Understanding terminate() and unexpected() 48 Setting the Terminate and Unexpected Handlers, The uncaught_exception () Function 49 Exception Handling 50 Formatting Using the ios Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 52 TB1 :20 TB1 :20 and Closing a File, Reading and Writing Text Files,	42		Functions, A Generic Sort, Compacting an Array		
43 Class, Using Non-Type Arguments with Generic Classes 43 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 44 Exception Handling: Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 45 Handling Derived-Class Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exception, Understanding terminate() and unexpected() 8% 81% 47 chapter Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 48 uncaught_exception() Function The exception and bad_exception Classes Applying Exception Handling 8% 81% 50 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 6% 87% 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own 6% 87% 52 TB1 :20 and Closing a File, Reading and Writing Text Files, 6% 87%			Generic Classes, An Example with Two Generic Data	6%	72%
43 Classes Classes 44 Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates 44 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 45 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions 46 TB1:19 47 Rethrowing an Exception, Understanding terminate() and unexpected() 48 Setting the Terminate and Unexpected Handlers, The uncaught_exception () Function 49 Exception Handling 50 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formattel I/O 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 52 Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,		chapter	Class Using Non-Type Arguments with Generic		
44 Explicit Class Specializations, The typename and export Keywords, The Power of Templates 45 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 46 TB1 :19 46 Handling Options, Catching All Exceptions, Restricting Exception, Understanding terminate() and unexpected() 47 chapter 48 Exception Andling 49 Exception Andling 49 Exception Handling 50 The exception Handling 50 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 52 TB1 :20 and Closing a File, Reading and Writing Text Files,	43		Classes		
44 export Keywords, The Power of Templates 45 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 46 TB1:19 47 chapter 48 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions 48 Nethrowing an Exception () Inderstanding terminate() and unexpected() 49 Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function 49 Exception Handling 50 Predefined Streams, Formatted I/O 50 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 51 Flags, Clearing Format Flags, Overloading << and >> 52 TB1:20 TB1:20 The IP (C): fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,			Using Default Arguments with Template Classes,		
45 Exception Handling: Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements 46 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions 47 Rethrowing an Exception, Understanding terminate() and unexpected() 48 Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function 49 Exception Handling 50 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s 50 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 52 TB1 :20 Affiel I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	44				
45 catch Statements 46 Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions, Restricting Exceptions, Restricting Exception, Understanding terminate() and unexpected() 8% 81% 47 chapter Rethrowing an Exception, Understanding terminate() and unexpected() 8% 81% 48 uncaugh_exception() Function Function 8% 81% 49 Exception Handling 8 81% 49 Exception Handling 8 81% 50 Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O 6% 87% 51 Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> 6% 87% 52 TB1 :20 TB1 i/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files, 6% 87%			Exception Handling: Exception Handling		· II
46Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions8%47rhapterRethrowing an Exception, Understanding terminate() and unexpected()8%48Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function The exception and bad_exception Classes Applying Exception Handling8%49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O8%50Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>51Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,6%	45				
46Handling Options, Catching All Exceptions, Restricting Exceptions8%47TB1 :19 chapterRethrowing an Exception, Understanding terminate() and unexpected()8%48Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function8%49Exception and bad_exception Classes Applying Exception Handling49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s50Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions52TB1 :20TB1 :20and Closing a File, Reading and Writing Text Files,	т Ј	-			
TB1 :19 chapterRethrowing an Exception, Understanding terminate() and unexpected()8%81%47Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function8%81%48The exception and bad_exception Classes Applying Exception Handling8%81%49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O6%50Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>6%51Extractors, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,87%			Handling Options, Catching All Exceptions,		
47ChapterRetinowing an Exception, Orderstanding terminate()8%81%47and unexpected()Setting the Terminate and Unexpected Handlers, The uncaugh1_exception() FunctionSetting the Terminate and Unexpected Handlers, The uncaugh1_exception() Function8%81%48The exception and bad_exception Classes ApplyingException Handling8%81%49Exception HandlingStandard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s8%87%50Predefined Streams, Formatted I/OFormatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>6%87%52TB1 :20TB1 :20AClosing a File, Reading and Writing Text Files,6%87%	46	TB1 ·19			
48 Setting the Terminate and Unexpected Handlers, The uncaught_exception() Function 48 Uncaught_exception () Function 49 Exception and bad_exception Classes Applying 49 Exception Handling 50 Standard C++ I/O Classes : Old vs. Modern C++ I/O, 50 Predefined Streams, The C++ Stream Classes, C++'s 50 Predefined Streams, Formatted I/O 51 Flags, Clearing Format Flags, Overloading << and >> 52 Extractors, Creating Your Own Inserters, Creating Your Own 52 TB1 :20 TB1 :20 and Closing a File, Reading and Writing Text Files,	47			8%	81%
48uncaught_exception() Function49The exception and bad_exception Classes Applying49Exception Handling49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s50Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>52Extractors, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	4/		1		
49The exception and bad_exception Classes Applying Exception Handling49Exception Handling49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s50Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>51Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	18		÷ ,		
49Exception Handling49Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s50Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>51Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	+0	4			
Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s50Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>51Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions52C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	49				
50C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions52TB1 :20TB1 :20Tile, Reading and Writing Text Files,			×		
50Predefined Streams, Formatted I/OFormatting Using the ios Members, Setting the Format51Formatting Using the ios Members, Setting the Format51Flags, Clearing Format Flags, Overloading << and >>Creating Your Own Inserters, Creating Your Own52Extractors, Creating Your Own Manipulator FunctionsC++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,					
51Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >> Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,6%	50				
51Flags, Clearing Format Flags, Overloading << and >>6%52Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,6%		1			
52Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,6%8/%	51			(0)	070/
52Extractors, Creating Your Own Manipulator FunctionsC++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,		1		٥%	8/%
C++ File I/O: fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files,	52		e e		
TB1 :20 and Closing a File, Reading and Writing Text Files,]			
53 Chapter Unformatted and Binary I/O		TB1 :20	ç ç		
	53	Chapter	Unformatted and Binary I/O		

54		STL : An Overview of the STL, Containers		
55		Algorithms, Iterators and Function Objects	6%	94%
56	TB1 :21	The String Class, Vector Class	070	9470
57	Chapter	Lists, Maps.		
58		Characters vs. Bytes, put() and get()		
59		read() and write(), More get()		
60		Functions, getline(), Detecting EOF		
		The ignore() Function, peek() and putback(), flush(),	8%	100%
61		Random Access		
62	TB1 :24 chapter	Obtaining the Current File Position , I/O Status, Customized I/O and Files.		

Syllabus for Sessionals :

Sessional #	Syllabus
T1	Class # 01 - 19
T2	Class # 20 -53
T3	Class # 54- 62

Literature

:

Publication info Book Author & Title Type Code Edition & Publisher ISBN # Text Herbert Schildt: C++ The Complete Tata McGraw Hill, TB1 ISBN=0071502394 Book Reference, 4th Edition, 2007. Text Book TB2 K R Venugopal, Rajkumar Buyya, T Ravishanker: Mastering C++ Tata McGraw Hill 10:0-07-463454-2 ISBN-13: 978-81-Al Stevens: C++ Programming, 7th Wiley India 265-References RB1 Edition, Publications 0398-8 Addison Wesley, ISBN 0-321-71411-Stanley B.Lippmann, Josee Lajore: References RB2 2005. 3 C++Primer, 4th Edition,

CMR Institute of Technology, Bangalore			All.
Department(s):Master of Computer Applications			CMR INSTITUTE OF TECHNOLOGY
Semester: 02	Section(s): A		
Database Management System		16MCA23	Lectures/week: 05
Course Instructor(s): Ms. Gomathi	Т		
Course duration: Feb 2017 – May 201	7		

Course Outcome:

By end of this course the students will be able to

- CO1: Demonstrate the fundamentals of data models and conceptualize and depict a database system and make use of ER diagram in developing ER Model.
- CO2: To summarize the SQL and relational database design.
- CO3: Illustrate transaction processing, concurrency control techniques and recovery.
- CO4: Inference the database design in the real world entities.

Class	Chapter Title /	Торіс	-	e of portion
#	Reference			ered
	Literature		Reference	Cumulative
		Characteristics of Database approach. Actors on the screen;		
1		Workers behind the scene		
		Advantages of using DBMS approach. Data models, schemas		
2		and instances		
3		Three-schema architecture and data independence.		
		Database Languages and interfaces, the database system		
4		environment, Centralized and Client server architecture		
5	Unit 1 :	Classification of Database Management System		
	INTRODUCTION	ER Model- Conceptual database using high level conceptual		
6		data models for database design.		
		A sample database applications, Entity types, Entity set		
7		attributes and keys		
8		Relationship types, Relationship sets		
9		Roles and structural constraints, Weak Entity type		
10		Exercise on ER Diagram		
11		Exercise on ER Diagram		
12		Exercise on ER Diagram		
13		Relational Model Concepts.		
		Relational Model Constraints and Relational Database		
14		Schemas.		
		Relational Model Constraints and Relational Database		
15		Schemas.		
		Update Operations, Transactions and dealing with constraint		
16	Unit 2 :	violations;		
	RELATIONAL	Update Operations, Transactions and dealing with constraint		
17	MODEL	violations;		
		Unary Relational Operations: SELECT and PROJECT;		
18		Relational Algebra Operations from Set Theory.		
19		Binary Relational Operations : JOIN and DIVISION		
20		Binary Relational Operations : JOIN and DIVISION		
21		Additional Relational Operations.		
22		Additional Relational Operations.		
23		Examples of Queries in Relational Algebra		
24		Examples of Queries in Relational Algebra		

			1	
25		Overview of the SQL Query Language, SQL Data Definition	4	
26		Basic structure of SQL Queries, Additional Basic Operations	-	
27	Unit 3 :	Null values, Aggregate functions, nested sub queries	-	
28	Introduction to	Modification of the database,	-	
29	SQL	Join Expression,	-	
30		Views and transactions		
31		Integrity constraints,		
32		SQL Data Types and Schemas ,Authorization		
33		SQL Data Types and Schemas ,Authorization		
34		Database programming issues and techniques		
35		Database programming issues and techniques		
36		Embedded SQL		
37		Informal Design Guidelines for Relation Schemas.		
38		Functional Dependencies.		
39		Functional Dependencies.		
40		Normal Forms Based on Primary Keys.		
41	Unit 4 :	Normal Forms Based on Primary Keys.		
42	Database Design	General Definitions of Second and Third Normal Forms.		
43		Boyce-Codd Normal Form.		
44		Boyce-Codd Normal Form.		
45		Stored Procedures and functions		
46		Stored Procedures and functions		
47		Triggers		
		Triggers		
48				
49		Transaction Concept	-	
50		A Simple transaction Model	-	
51	Unit 5 :	Transaction atomicity and Durability	-	
52		Serializability	-	
53	Transaction	Transaction isolation levels	-	
54	Management	Implementation of isolation levels	-	
55		Concurrency control	-	
56		Lock based protocols	-	
57		Deadlock handling	-	
58		Recovery System: Failure Classification		
59		Storage]	
60		Recovery and Atomicity		
61		Recovery algorithm		
62		Recovery algorithm]	

Syllabus for Sessionals:

Sessional #	Syllabus
T1	Class #1 -17 ,Class # 26- 40
Т2	Class # 18 -25, Class # 41
12	-62

Literature:

Book Type	Code	Author & Title	Publicati	on info
			Edition & Publisher	ISBN #
Text Book	TB1	Elmasri and Navathe, "Fundamentals of Database Systems".	5 th edition, Pearson Education	978-81-317-1625-0
Text Book	TB2	Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems.	3 rd Edition, McGraw- Hill, 2003.	0-07-123151-X
Reference Book	RB2	Silberschatz, Korth and Sudharshan: Data base System Concepts.	5 th Edition, Mc- GrawHill ,2006	0-07-295886-3
References	RB2	Database Principles Fundamentals of Design, Implementation and Management by Coronel, Morris	Rob-Cengage Learning,2012	978-1-133-31197-3

CMR Institute of T	echnology, Bangalore		9112
Department(s):Ma	aster of Computer Applications	;	
Semester: 02	Section(s): A		
Operating System	ns	10MCA24	Lectures/week: 05
Course Instructor	s): Ms. Helen Josephine VL		
Course duration: I	eb 2017 – June 2017		

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

- CO1: Understand the Basics of Computer and Operating Systems Structure
- CO2: Realize the concept of Process Management and Mutual Execution
- CO3: Understand the concepts of the Deadlock and different approaches to memory management.
- CO4: Learn the concepts of file system
- CO5: Understand the concepts of Computer Security.

Class #	Chapter Title / Reference	Торіс	Percentage of portion covered	
	Literature		Reference	Cumulative
1	TB2:1.1 TO 1.7 TB1: 1.1 TO 1.10 TB1:2.1 TO 2.6	Computer and Operating Systems Structure: Basic Elements, Processor Registers, Instruction Execution	17%	17%
2		The Memory Hierarchy, Cache Memory, I/O Communication Techniques, Introduction to Operating		

]	System]	
3	-	Mainframe Systems, Desktop System	_	
4	-	Multiprocessor Systems, Distributed Systems,	-	
		Clustered Systems,		
5		Real - Time Systems, Handheld Systems		
6		Feature Migration, Computing Environments.	_	
7		System Components, Operating – System Services	_	
8		System Calls, System Programs	-	
9		System Structure, Virtual Machines	-	
10	-	System Design and Implementation	-	
11	-	System Generation	-	
12		Process Management and Mutual Execution:		
		Process, Process States		
13	-	Process Description, Process Control	_	
14	-	Execution of the Operating System	-	
15	-	Security Issues, Processes and Threads	-	
16	-	Symmetric Multiprocessing(SMP), Micro kernels	-	
17	TB2:3.1 TO 3.04	CPU Scheduler and Scheduling	19%	36%
18	TB2:4.1 TO 4.3 TB2:5.1 TO 5.6	Principles of Concurrency	-	
19	-	Mutual Exclusion: Hardware Support	-	
20	-	Semaphores	-	
21	-	Monitors	_	
22	-	Message Passing	_	
23	-	Readers/Writes Problem	_	
24		Deadlock and Starvation :		
	TB2:6.1 TO 6.6	Principles of Deadlock		
25	TB2: 7.1 TO 7.5 TB1:10.1 TO	Deadlock Prevention, Deadlock Avoidance,	20%	56%
26	10.6	Deadlock Detection, An Integrated Deadlock Strategy	-	
27	-	Dining Philosophers Problem	-	
	1		_	I

29Contiguous Memory Allocation30Paging, Segmentation31Segmentation with Paging32Demand Paging, Examples33Process Creation, Examples34Page Replacement, Examples35Allocation of Frames, Thrashing with Examples36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting42File – System Mounting4312.5TB:: 14.1 TOFile – System Structure44File – System Structure45File – System Structure46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Structure51TB:: 19.1 TO52TB:: 19.1 TO53TB:: 20.1 TO54Program Threats55System Threats	28		Memory Management :Swapping		
31 Segmentation with Paging 32 Segmentation with Paging 33 Demand Paging, Examples 34 Page Replacement, Examples 35 Allocation of Frames, Thrashing with Examples 36 File System and secondary storage: File concept 37 Access Methods 38 Directory Structure, File – System Mounting, 40 File - System Mounting 41 File - System Mounting 42 File - System Mounting 43 File - System Mounting 44 File - System Mounting 45 File - System Implementation 46 Directory Implementation 47 Allocation Methods 48 Free -Space Management 49 Disk Structure 50 Disk Structure 51 Computer Security Problem 52 TB1: 19.1 TO 53 TB1: 20.1 TO 53 TB2: 20.1 TO 53 Structure Problem 54 Vergram Threats	29		Contiguous Memory Allocation		
32Demand Paging, Examples33Demand Paging, Examples34Page Replacement, Examples35Allocation of Frames, Thrashing with Examples36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,39Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting42File – System Mounting43T2.5TB1: 12.1 TO43T2.5TB2: 14.1 TO4414.245File – System Structure46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Structure51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 20.1 TO 20.953Forgram Threats	30		Paging, Segmentation		
33Process Creation, Examples34Page Replacement, Examples35Allocation of Frames, Thrashing with Examples36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,39Directory Structure, File – System Mounting,40File – System Mounting41TB1: 11.1 TO42TB1: 21.2 TO11.6File – System Mounting4312.5TB1: 14.1 TO4414.245File – System Structure46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Structure51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 20.1 TO 20.953Program Threats	31		Segmentation with Paging		
34Page Replacement, Examples35Allocation of Frames, Thrashing with Examples36Allocation of Frames, Thrashing with Examples36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting41File – System Mounting41File – System Mounting43TBI: 11.1 TO12.5Protection,7BI: 14.1 TOFile – System Structure44File – System Implementation45Directory Implementation46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Scheduling51TBI: 19.1 TO 19.452TBI: 20.1 TO 20.953The Security Problem53The Security Problem54Variant Science Study of Linux Operating system : The Security Problem53Sa Authentication54Program Threats	32		Demand Paging, Examples		
35Allocation of Frames,Thrashing with Examples36Allocation of Frames,Thrashing with Examples36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting42File – System Mounting4312.5TB1: 12.1 TOFile – System Mounting4414.245File – System Structure46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.453The Security Problem53The Security Problem54Program Threats	33		Process Creation, Examples		
36File System and secondary storage: File concept37Access Methods38Directory Structure, File – System Mounting,39Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting41File – System Mounting41File – System Mounting43File – System Mounting44File – System Mounting44File – System Mounting44File – System Mounting45File – System Structure46File – System Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Structure51TB1: 19.1 TO 19.452TB1: 19.1 TO 19.453TB1: 20.1 TO 20.954Program Threats	34		Page Replacement, Examples		
37Access Methods38Directory Structure, File – System Mounting,39Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting41File – System Mounting41File – System Mounting42TB1: 12.1 TO4312.5 TB1: 14.1 TO4414.245File – System Structure44File – System Implementation46Directory Implementation47Allocation Methods48Free –Space Management49Disk Structure50Disk Structure51TB1: 19.1 TO 19.452TB1: 20.1 TO 20.954The Security Problem5320.954Program Threats	35		Allocation of Frames, Thrashing with Examples		
38Directory Structure, File – System Mounting,39Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting41File – System Mounting41File – System Mounting42TB1: 12.1 TO4312.5TB1: 14.1 TOProtection,44File – System Structure45File – System Implementation46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.453Computer Security Problem54The Security Problem	36		File System and secondary storage: File concept		
39Directory Structure, File – System Mounting,40File – System Mounting41File – System Mounting42TB1: 11.1 TO 11.6 TB1: 12.1 TO 12.25 181: 14.1 TO43TB1: 12.1 TO 14.244File – System Mounting44File – System Structure45File – System Implementation46Directory Implementation47Allocation Methods48Free – Space Management49Disk Structure50Disk Structure51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.4 TB1: 20.1 TO 20.953TB1: 20.1 TO 20.954Program Threats	37		Access Methods	-	
40File - System Mounting41File - System Mounting42TB1: 11.1 TO 11.643T2.5TB1: 12.1 TO 12.544T4.245File - System Structure46File - System Implementation47Allocation Methods48Free -Space Management49Disk Structure50TB1: 19.1 TO 19.451Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 20.1 TO 20.953TB1: 20.1 TO 20.954Program Threats	38		Directory Structure, File – System Mounting,	-	
41 TB1: 11.1 TO 11.6 File - System Mounting 24% 80% 42 TB1: 12.1 TO 12.5 File Sharing, 24% 80% 43 T2.5 File - System Structure 24% 80% 44 14.2 File - System Structure 24% 80% 45 File - System Implementation Directory Implementation 24% 80% 46 Allocation Methods Free -Space Management Disk Structure 50 51 Computer Security and Case study of Linux Operating system : The Security Problem 20 100 51 TB1: 19.1 TO 19.4 The Security Problem 20 100 52 TB1: 19.1 TO 19.4 The Security Problem 20 100 53 T81: 20.1 TO 20.9 Viser Authentication 20 100	39		Directory Structure, File – System Mounting,	-	
TB1: 11.1 TO 11.6 TB1: 12.1 TO 12.5 TB1: 14.1 TOFile Sharing,4312.5 TB1: 14.1 TOProtection,24%80%4414.2File – System Structure14.2File – System Implementation24%80%4614.2File – System ImplementationDirectory Implementation14.114.214.214.214.247Allocation MethodsFree – Space ManagementDisk Structure14.214.214.214.214.250Disk StructureDisk Structure14.214.214.214.214.214.250Directory ImplementationDirectory Implementation14.214.214.214.250Disk StructureDisk Structure15.210.210.251TB1: 19.1 TO 19.4 TB1: 20.1 TO 20.9The Security Problem2010052TB1: 20.1 TO 20.9The Security Problem201005354Program Threats20100	40		File – System Mounting	-	
4211.6 TB1: 12.1 TO 12.5 TB1: 14.1 TOFile Sharing,4312.5 TB1: 14.1 TOProtection,24%80%4414.2File – System StructureFile – System Implementation24%80%4514.2File – System ImplementationDirectory Implementation24%80%46Allocation MethodsFree –Space ManagementDisk Structure10048Free –Space ManagementDisk Structure10010050Disk Structure10010010051Computer Security and Case study of Linux Operating system : The Security Problem2010052TB1: 20.1 TO 20.9The Security Problem201005320100	41	TB1· 11 1 TO	File – System Mounting	-	
4312.5 TB1: 14.1 TO 14.2Protection, File - System Structure24%80%4414.2File - System StructureFile - System Implementation24%80%45Directory ImplementationDirectory ImplementationAllocation Methods14.214.246Allocation MethodsFree -Space Management10 isk Structure14.210 isk Structure48Disk StructureDisk Structure15 isk Scheduling15 isk Scheduling20 isk Structure50TB1: 19.1 TO 19.4The Security and Case study of Linux Operating system : The Security Problem20 isk Scheduling20 isk Scheduling51TB1: 20.1 TO 20.9User Authentication20 isk Scheduling20 isk Scheduling53TB1: 20.1 TO 20.9User Authentication20 isk Scheduling54Program Threats100 isk Scheduling	42	11.6	File Sharing,		
41File - System Implementation45File - System Implementation46Directory Implementation47Allocation Methods48Free - Space Management49Disk Structure50Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.453The Security Problem54Viser Authentication	43	12.5	Protection,	24%	80%
46Directory Implementation47Allocation Methods48Free -Space Management49Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.453TB1: 20.1 TO 20.954Program Threats	44	14.2	File – System Structure		
47Allocation Methods48Free –Space Management49Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.4 TB1: 20.1 TO 20.953Z0.954Program Threats	45		File – System Implementation	•	
48Free -Space Management49Disk Structure50Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.4 TB1: 20.1 TO 20.953TB1: 20.1 TO 20.954Program Threats	46		Directory Implementation		
49Disk Structure50Disk Scheduling51Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.4 TB1: 20.1 TO 20.953TB1: 20.1 TO 20.954Viser Authentication54Program Threats	47		Allocation Methods		
50Disk Scheduling51Disk Scheduling51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.452TB1: 20.1 TO 20.953TB1: 20.1 TO 20.954Program Threats	48		Free –Space Management	•	
51Computer Security and Case study of Linux Operating system : The Security Problem52TB1: 19.1 TO 19.4The Security Problem53TB1: 20.1 TO 20.9User Authentication54Program Threats20	49		Disk Structure	-	
System : The Security Problem52TB1: 19.1 TO 19.453TB1: 20.1 TO 20.954User Authentication54Program Threats	50		Disk Scheduling		
5219.4The Security Problem2010053TB1: 20.1 TO 20.9User Authentication2010054Program Threats000	51				
53 20.9 User Authentication 54 Program Threats	52	19.4	The Security Problem		
	53		User Authentication	20	100
55 System Threats	54		Program Threats		
	55		System Threats		

56		Linux System Linux history	
57		Design Principles	
58		Kernel modules	
59		Process, management	
60	•	Scheduling, Memory management	
61		File systems, Input and output	
62		Inter-process communications	

Syllabus for Sessionals :

Sessional #	Syllabus
T1	#1 - #30
T2	#31 - #62

Literature:

Book Type	Code	Author & Title	Publication info	
			Edition & Publisher	ISBN #
Text Book	TB1	Silberschatz, Galvin, Gagne, "Operating System Concepts" John Wiley,	Sixth Edition, 2004	978-0471417439
Text Book	TB2	William Stallings, "Operating Systems – Internals and Design Principles" Pearson,	6 th edition, 2012	10: 0133805913
References	RB1	Chakraborty , "Operating Systems" Jaico Publishing House	, 2011.	9788179929766
References	RB2	Dhananjay M. Dhamdhere, "Operating Systems – A Concept – Based Approach", Tata McGraw – Hill,	3rd Edition, 2012	9781259005589

CMR Institute of T			
Department(s).wa	aster of Computer Application	5115	
Semester: 02	Section(s): A		i.c.indeodi
System Software		10MCA25	Lectures/week: 05
Course Instructor	s): Ms. Ashwini Patil		
Course duration: I	eb 2017 – June 2017		

Course Objective:

To view some of the major tasks of the system software of a computer system, focusing on internal working of the hardware and software interface of a typical system..

Course Pre-requisites:

Student should have basic knowledge of mnemonic codes.

Course Outcome:

CO1: Understand the introductory concepts of system software, SIC and SIC/XE machine

architecture.

CO2: Understand the design and implementation of Assemblers with implementation examples.

CO3: Design and implement the linkers and loaders, macro processors and respective

implementation examples.

CO4: Learn the basic design and working of compilers.

Class #	Chapter #	Topics to be covered	% of porti	ions covered
			Reference Chapter	Cumulative
1 2 3 4 5 6 7	Unit –1 Machine Architecture & Introduction to Assemblers	Introduction to System Software & Machine architecture.SIC/standard version -Machine Architecture – Memory, Registers, Data Formats, Instruction format, Addressing Modes, Instruction Set, Input and Output.SIC/XE –Machine Architecture – Memory, Registers, Data Formats, Instruction format, Addressing Modes, Instruction Set, Input and Output.SIC – Programming Examples (Data movement, arithmetic operation & Indexing, looping)Traditional CISC Machines-VAX ArchitectureRISC Machines-UltraSPARC ArchitectureBasic Assembler Functions A Simple SIC Assembler	Chapter 15.00	15.00
8		Assembler algorithm and Data Structure		
9		Pass I Algorithm		
10		Pass II Algorithm		
11		Machine-Dependent Assembler Features - Instructions Formats and Addressing Modes	25.00	40.00

12	Unit – 2 Assembler	Program Relocation criteria		
13		Machine-Independent Assembler Features- Literals		
14		Symbol Defining Statements		
15		Expressions		
16		Program Blocks		
17		ControlSection and Program Linking		
18		Assembler Design Options One-Pass Assemblers		
19		Multi Pass - Assemblers		
20		Implementation Example-MASM assembler		
21				
		SPARC assembler		
22		Basic Loader Functions Design of an Absolute Loader		
23		Simple Bootstrap Loader		
24		Machine-Dependent Loader Features- Relocation		
25		Program Linking		
26		Algorithm and Data Structures for Linking Loader		
27	Unit 3-	Machine-Independent Loader Features - Automatic Library Search		
28	Loaders and Linkers	Loader Options	20.00	60.00
29		Loader Design options- Linkage Editors		
30		Dynamic Linking		
31		Bootstrap Loader		
32		Implementation Example-MSDOS Linker		
33		SunOS Linker		
34		Cary MPP Linker		
35		Basic Macro Processors Functions Macro Definition and Expansion		
36		Macro Processors – Data Structures		
37	Unit-4 Macro Processor	Algorithm for a One – Pass Macro Processor	20.00	80.00
		Machine Independent Macro Processor features - Concatenation of Macro	20.00	00.00
38		Parameters, Generation of Unique Labels		
39		Conditional Macro Expansion		

40		Keyword Macro Parameters		
41		Recursive Macro Expansion		
42		General- Purpose Macro Processors		
		Macro Processing Within Language		
43		Translators		
44		Implementation Examples- MASM Macro Processor		
45		ANSIC Macro Language		
46				
		ELENA macro processor		
47				
		Basic Compilers Functions- Grammars		
48		Lexical Analysis		
49		Syntactic Analysis		
50		Code Generation		
51		Machine Dependent Compiler Features- Intermediate Form of the Program		
52		Machine dependent code		
		Optimization		
53	Unit-5	Machine Independent Compiler		
	Compilers	Features- Structured variables	20.00	100
54		Machine Independent code Optimization	20.00	100
55		Compiler Design Options- Division into		
		passes		
56		Interpreters		
57		P-code Compilers		
58		Compiler-Compilers		
59		SunOS C compiler,		
60		YACC Compiler-compiler.		

Syllabus for Sessionals :

Sessional #	Syllabus
T1	Class # 01 – 21
T2	Class # 21 – 40
T3	Class # 41 – 60

Literature:

Book Type	Code	Author & Title
Text Book	TB1	Leland.L.Beck: System Software, 3rd Edition, Addison-Wesley, 1997.
References	RB1	J.Nithyashri, "System Software", 2nd Edition, Tata McGraw Hill, 2010