**Master of Computer Applications**

**THIRD SEMESTER EXAMINATION**

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Paper Code</th>
<th>Paper</th>
<th>L</th>
<th>T/P</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>044201</td>
<td>MCA 201</td>
<td>Theory of Computation</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>044203</td>
<td>MCA 203</td>
<td>Computer Graphics</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>044205</td>
<td>MCA 205</td>
<td>Java Programming</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>044207</td>
<td>MCA 207</td>
<td>Data Communications and Networking</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>044209</td>
<td>MCA 209</td>
<td>C# Programming</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

**Practical**

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Paper Code</th>
<th>Paper</th>
<th>L</th>
<th>T/P</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>044251</td>
<td>MCA 251</td>
<td>Computer Graphics Lab</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>044253</td>
<td>MCA 253</td>
<td>Java Programming Lab</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>044255</td>
<td>MCA 255</td>
<td>C# Programming Lab</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

**NUES**

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Paper Code</th>
<th>Paper</th>
<th>L</th>
<th>T/P</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>044261</td>
<td>MCA 261</td>
<td>General Proficiency – III*</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(It is suggested to have Technical Paper Writing Course)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total**

|         |             |                                           | 15 | 17  | 26     |

*Non-University Examination System (NUES)*
**OBJECTIVES:**
This course is extensive and theoretical treatment of issues in Computability and Complexity; Topics include Automata and Language Theory, Computability Theory, and Complexity Theory. Learning outcome of this course will be theoretical treatment of following
- What can be computed and how fast it can be done?
- Use of Automata and Language theory in the development of different modules of a compiler as a case study.

**PRE-REQUISITE:**
- Discrete Mathematics
- Skills in writing Formal Mathematical Proofs

**UNIT - I**
Automata and Language Theory: Overview of Theoretical Computer Science (including computationally intractable problems), Introduction to System software including various phases/Modules in the design of a typical compiler, Chomsky Classification, Finite Automata, Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), statement of Kleen's Theorem, Regular Expressions, Equivalence of DFAs, NFAs and Regular Expressions, Closure properties of Regular Language, Non-Regular Languages, Pumping Lemma, Myhill Nerode Theorem, Use of Regular expressions in the Design of scanner (lexical analyzer). Introduction to JFLAP Simulation.

[No. of Hr: 12]

**UNIT - II**
Context Free Languages: Context Free Grammar (CFG), Parse Trees, Push Down Automata (deterministic and nondeterministic) (PDA), Equivalence of CFGs and PDAs, Closure properties of CFLs, Pumping Lemma, Parsing (including LL(1), SLR and LR(1) Parsing Method).

[No. of Hr.: 12]

**UNIT - III**
Turing Machines and Computability Theory: Definition of Turing Machine, Extensions of Turing machines, Non-deterministic Turing machines, Equivalence of various Turing Machine Formalisms, Church – Turing Thesis, Decidability, Halting Problem, Reducibility, Recursion Theorem.

[No. of Hr: 10]

**UNIT - IV**
Complexity Theory: Time and Space measures, Hierarchy theorems, Complexity classes P, NP, space complexity, Savich theorem, L, NL, PSPACE complexity, Post correspondence problem, Probabilistic computation.

[No. of Hr: 6]

**TEXT BOOKS:**

REFERENCES:
OBJECTIVES: Computer graphics is used in diverse applications from the visualization of complex scientific data to the special effects in computer games. The objective of this course is to introduce the programming principles of computer graphics. The course will cover Practical programming through C, and mathematical and theoretical foundations.

PRE-REQUISITE:
- Mathematical Concepts

UNIT-I
Scan conversion: Scan converting a point, line (Bresenham’s, DDA), 2-D transformations (Rotation, Rotation about an arbitrary line, Scaling, Translation, Shearing, Reflection, and Reflection about an arbitrary line), circle and ellipse.
Transformation: 2D transformation, Basic Transformation, Various 2D and 3D Transformation matrices (Translation, Rotation, Scaling, Shearing and Reflection), Composite transformations: Reflection, Shearing and Transformation between coordinate Systems. Rotation about : (i) an arbitrary axis (ii) about an arbitrary point. 

UNIT-II
Curves and Surfaces Bezier Curves, 4 point and 5 point Bezier curves using Bernstein Polynomials, Conditions for smoothly joining curve segments, Bezier bi-cubic surface patch, B-Spline Curves, Cubic B-Spline curves using uniform knot vectors, first and second order continuities, Effect of multiple control points at same location, Geometrical Construction, Computing control points given end slopes for a specified curve segment.

UNIT-III
Transformations: 3-D Transformation, Computing location of V.P, 2-D viewing, Window-to-view port transformation
Clipping: Line Clipping; Sutherland Cohen clipping algorithms, Sutherland-Hodgeman.
Projection: Parallel and Perspective Projections
Solid Modeling: Sweeping a polygon or a surface patch along a path to form solids, Boundary Representation (B-Rep), octrees, CSG – Constructive Solid Geometry.

UNIT-IV

INSTRUCTIONS TO PAPER SETTERS:
1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.
TEXT BOOKS:

REFERENCES:
INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVE: In this course student will become familiar with features of Java language, they will learn how to write Java code according to Object-Oriented Programming principles, how to design GUI applications and Applets using AWT, how to develop multithreaded and Networking applications and how to create dynamic pages.

PRE-REQUISITES:
- Basic Object Oriented Programming Concepts

UNIT – I
Importance and features of Java, Language Construct of java including Keywords, constants, variables and looping and decision making construct, Classes and their implementation, Introduction to JVM and its architecture including set of instructions. Overview of JVM Programming. Internal and detailed explanation of a valid .class file format. Instrumentation of a .class file, Byte code engineering libraries, Overview of class loaders and Sandbox model of security.
Introducing classes, objects and methods: defining a class, adding variables and methods, creating objects, constructors, class inheritance. Arrays and String: Creating an array, one and two dimensional arrays, string array and methods, Classes: String and String Buffer classes, Wrapper classes: Basics types, using super, Multilevel hierarchy abstract and final classes, Object class, Packages and interfaces, Access protection, Extending Interfaces, packages.

[No. of Hrs.: 12]

UNIT – II
Exception Handling: Fundamentals exception types, uncaught exceptions, throw, throw, final, built in exception, creating your own exceptions,
Multithreaded Programming: Fundamentals, Java thread model: priorities, synchronization, messaging, thread classes, Runnable interface, inter thread Communication, suspending, resuming and stopping threads.
Input/Output Programming: Basics, Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files.

[No. of Hrs.: 10]

UNIT – III
Event Handling: Different Mechanism, the Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and Inner Classes, Working with windows, Graphics and Text, using AWT controls, Layout managers and menus, handling Image, animation, sound and video, Java Applet.
The Collection Framework: The Collection Interface, Collection Classes, Working with Maps & Sets

[No. of Hrs: 09]

UNIT – IV
Java Bean: Introduction, Bean Architecture, Using the Bean Development Kit, Creating simple bean-properties, methods and events, Packing beans- the manifest & the jar, Java bean package, Introduction to NetBean.
Swing: Introduction to JFC (Java Foundation Classes), Features of Swing, Comparison with AWT, Advanced Control.

[No. of Hrs.: 11]

TEXT BOOKS:

REFERENCES:
OBJECTIVE:
This course covers theory and practice of data communication between computing devices. Topics include network architecture and topology, Basics of networking and protocols, OSI network layered models and Application layer protocols.

PRE-REQUISITE:
- Basic Networking
- Operating System Concepts

UNIT - I
Introductory Concepts: Goals and Applications of Networks, Network structure and architecture, the OSI reference model, services, networks topology.

UNIT - II

UNIT - III
Network Layer: Point-to-Point network, routing algorithms, congestion control, internetworking, Quality Control, Internetworking, The Network Layer in the Internet, IP packet, IP addresses, IPv6. [No of Hrs.: 10]

UNIT - IV
TEXT BOOKS:
4. Comer, “Computer Networks and Internet”, PHI.
5. Comer, “Internetworking with TCP/IP”, PHI.

REFERENCES:
INSTRUCTIONS TO PAPER SETTERS:
1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions
   of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every
   unit should have two questions to evaluate analytical/technical skills of candidate. However,
   student may be asked to attempt only 1 question from each unit. Each question should be 10 marks
   including subparts, if any.

OBJECTIVE: In this course student will become familiar with an with C# language. This course will
help to develop real life projects.

PREREQUISITES:
- Basic Programming Language

UNIT - I
The CLR and .NET Framework: Understand the motivation behind the .NET platform, Common
Language Infrastructure (CLI). Know the role of the Common Type System (CTS), the Common
Language Specification (CLS) and the Common Language Runtime (CLR), Understand the assembly,
metadata, namespace, type distinction, Contrast single-file and multi-file assemblies, Know the role of
the Common Intermediate Language (CIL), Platform independent .NET(Mono / Portable .NET
distributions).
[No. of Hrs.: 08]

UNIT - II
Evolution of C# Language: Language Fundamentals, Reference and value Types, primitive types the
Nullable and enum types, Classes and objects, Defining classes Creating objects, Using static
members, Garbage Collector, Overloading Methods, Various Constructors. Encapsulating data, access
modifiers, properties, indexers arrays and readonly fields. Handling errors and throwing exceptions The
Root object class. Inheritance and polymorphism specialization and generalization, Abstract classes,
esting of classes. Structures. String and DateTime classes.
[No. of Hrs: 14]

UNIT - III
Event handling paradigm Delegates and events. Anonymous delegates and lambda expression
FUNC and Action delegates.
Generics Collections Interfaces, overriding interface implementation. Explicit interface implementation.
Collection, IEnumerable, IEnumerator, IList, IComparer and their Generic equivalent. Working with
generic List, Stack, Dictionary and Queue.
Programming Window Forms Applications: The notifies - subscribers paradigm for handling events.
.NET framework for handling GUI events. Introduction to WPF and building an WPF application
[No. of Hrs: 10]

UNIT - IV
Introducing LINQ and XML: XML A quick introduction. LINQ and C#. Defining and executing a
Query. Implicitly typed local variables. Anonymous Types, Extension Methods and Lambda Expressions.
Putting LINQ to work. LINQ to SQL Fundamentals of ADO.NET Updating retrieving and deleting data
using LINQ to SQL. [No. of Hrs: 10]
TEXT BOOKS:
1. Jesse Liberty and Donald Xie, “Programming C# 3.0”, O’REILLY.
4. Joseph Albahari and Ben Albhari, “C# 3.0/4.0 in NUTSHELL”, O’REILLY.

REFERENCES:
2. Jon Skeet, “C# in Depth”, O’REILLY
Practical will be based on following:

1. Computer Graphics Lab \hspace{0.5cm} MCA 211
2. Java Programming Lab \hspace{0.5cm} MCA 213
3. C# Lab \hspace{0.5cm} MCA 215
Code No. : MCA 261
Paper: General Proficiency – III*

It is suggested to have a fundamental course on Technical Paper Writing in this semester.

This paper is under Non University Examination system its detail content will be decided by the respective Institute, under approval of the coordination committee based on the requirement of individual institution.

*Non University Examination Scheme (NUES)

There will not be any external examination of the university. The performance of the candidates should continuously be evaluated by an internal committee. The committee may conduct viva-voce at the end for the award of the marks.