UNIVERSITY OF KERALA

B. TECH. DEGREE COURSE
(2013 SCHEME)

SYLLABUS FOR
V SEMESTER
INFORMATION TECHNOLOGY
<table>
<thead>
<tr>
<th>Course No</th>
<th>Name of subject</th>
<th>Credits</th>
<th>Weekly load, hours</th>
<th>C A Marks</th>
<th>Exam Duration Hrs</th>
<th>U E Max Marks</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.501</td>
<td>Engineering Mathematics IV (AFRT)</td>
<td>4</td>
<td>3 3 1</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>(Complex Analysis and Linear Algebra)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.502</td>
<td>Engineering Mathematics- V (FR)</td>
<td>4</td>
<td>3 3 1</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>(Advanced Mathematics and Queueing Models)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.503</td>
<td>Operating Systems ( FR)</td>
<td>3</td>
<td>2 1</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>13.504</td>
<td>Systems Programming ( FR)</td>
<td>3</td>
<td>2 1</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>13.505</td>
<td>Theory of Computation (F)</td>
<td>4</td>
<td>2 2</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>13.506</td>
<td>Software Architecture &amp; Design Patterns ( F)</td>
<td>3</td>
<td>2 1</td>
<td>50</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>13.507</td>
<td>Digital Circuit Lab (F)</td>
<td>4</td>
<td>- -</td>
<td>4</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>13.508</td>
<td>Database Lab( F)</td>
<td>4</td>
<td>- -</td>
<td>4</td>
<td>3</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>29</td>
<td>14 7 8</td>
<td>400</td>
<td>800</td>
<td>1200</td>
<td></td>
</tr>
</tbody>
</table>

**SCHEME -2013**

**V SEMESTER**

**INFORMATION TECHNOLOGY (F)**
13.501 ENGINEERING MATHEMATICS – IV (AFRT)
(COMPLEX ANALYSIS AND LINEAR ALGEBRA)

Teaching Scheme: 3(L) - 1(T) - 0(P) 
Credits: 4

Course Objective:

- To introduce the basic notion in complex analysis such as Analytic Functions, Harmonic functions and their applications in fluid mechanics and differentiations and integration of complex functions, transformations and their applications in engineering fields.

- Many fundamental ideas of Linear Algebra are introduced as a part of this course. Linear transformations provide a dynamic and graphical view of matrix-vector multiplication. Orthogonality plays an important role in computer calculations.

Module – I


Conformal mapping: Conformality and properties of the transformations \( w = \frac{1}{z} \), \( w = z^2 \), \( w = z + \frac{1}{z} \), \( w = \sin z \), \( w = e^z \) - Bilinear transformations.

Module – II


Evaluation of real definite integrals – \( \int_0^{2\pi} f(\sin x, \cos x)dx \), \( \int_{-\infty}^{\infty} f(x)dx \) (with no poles on the real axis). (Proof of theorems not required).

Module – III

Vector spaces and subspaces - Null spaces, Column spaces and linear transformations-Kernal and range of a linear transformation -Linearly independent sets-Bases –Bases for nulA and ColA-Co-ordinate systems -Dimension of vector space -Rank -Change of basis.

Module – IV

Inner product spaces -Length and orthogonality -Orthogonal sets-Orthogonal and orthonormal bases -Orthogonal projection -Gram-Schmidt process -Least square problem -Quadratic forms- Constrained optimization of quadratic forms -Singular value decomposition (proof of the theorem are not included).
References:


**Internal Continuous Assessment (Maximum Marks-50)**

50% - Tests (minimum 2)
30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.
20% - Regularity in the class

**University Examination Pattern:**

Examination duration: 3 hours Maximum Total Marks: 100

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question out of the two from each module. Each question carries 20 marks.

**Course Outcome:**

After successful completion of this course, the students master the basic concepts of complex analysis and linear algebra which they can use later in their career.
Course Objectives:

- Mathematical programming techniques are introduced as a part of this course. These techniques are concerned with the allotment of available resources so as to minimize cost or maximize profit subject to prescribed restrictions.

- The study of queueing models provides us methods to minimize the sum of cost of providing service and cost of obtaining service which are primarily associated with the value of time spent by the customer in a queue.

- Network models such as PERT and CPM are introduced as a part of this course which are used for planning, scheduling and controlling complex projects.

Module – I


Module – II


Module – III

Network Analysis-Project scheduling –construction of project networks-critical path method(CPM)-Identification of critical path using CPM-Estimation of Floats-Total float-Free float-Independent float-Project evaluation and review technique (PERT)-Computation of expected completion times by PERT.

Module – IV

Queueing Theory-Queues –Characteristic of queues-Random arrivals-Arrival and Departure Distributions-Types of queues-Little’s Formulae-Basic queueing models – M/M/1:∞/FIFO,M/M/C:∞/FIFO,M/M/I:K/FIFO-Basic queue characteristics of models.

References


**Internal Continuous Assessment (Maximum Marks-50)**

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as home work, problem solving, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

**University Examination Pattern:**

Examination duration: 3 hours  
Maximum Total Marks: 100

The question paper shall consist of 2 parts.

**Part A (20 marks)** - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

**Part B (80 Marks)** - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

**Course outcome:**

At the end of the course, the student will be familiar with large scale applications of operations research techniques computations which require only few minutes on the computer.
Course Objectives:

To provide an understanding of concepts those underlie operating systems.

Module – I


Module – II


Module – III


Module – IV

Dead locks: Dead lock problem - characteristics - prevention - avoidance - detection - Recovery from dead lock - combined approach to dead lock handling. Protection: Goals of protection sms and policies - domain of protection - access matrix and its implementation. Dynamic protection structures, security.

References:


**Internal Continuous Assessment (Maximum Marks-50)**

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

**University Examination Pattern:**

Examination duration: 3 hours  Maximum Total Marks: 100

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain subdivisions), out of the two from each module. Each question carries 20 marks.

**Note:** The question paper shall contain at least 50% analytical/problem solving questions.

**Course Outcome:**

After successful completion of this course, the student will be able to understand how operating system works in the background and makes the user interact with the machine.
13.504 SYSTEM PROGRAMMING (FR)

Teaching Scheme: 2(L) - 1(T) - 0(P)  
Credits: 3

Course Objective:

- To impart the basic concepts of system software design.
- To equip the student with the right kind of tools for computer systems design and development.

Pre-requisites:  
13.402 - Computer Organisation and Design  
13.306 - Data Structures and Algorithms.

Module – I


Module – II


Module – III


Macro processors – Basic macro processor functions, machine dependent and machine independent macro processor features, Design options. Macro implementation – MASM, ANSI C macro processors.

Module – IV

Text Editors – overview of the editing process, user interface, editor structure. Debuggers – Overview of Debugger features, Breakpoint mechanism, Hardware support for debugging, Context of Debugger Check pointing and Reverse Execution.

General overview of the UNIX operating system - history of UNIX - system structure - user perspective - services - hardware assumptions - unix architecture - system concepts - kernel data structures - system administration process (concepts only).
References:


**Internal Continuous Assessment (Maximum Marks-50)**

- **50% - Tests (minimum 2)**
- **30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project etc.**
- **20% - Regularity in the class**

**University Examination Pattern:**

*Examination duration: 3 hours*  
*Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

**Part A (20 marks)** - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

**Part B (80 Marks)** - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

**Note:** The question paper shall contain at least 30% analytical/problem solving questions.

**Course Outcome:**

After the successful completion of the course students will be able to:

- Design and develop various system softwares.
- Take more advanced software courses.
- Self learn advance features in system softwares.
13.505 THEORY OF COMPUTATION (F)

Teaching Scheme: 2(L) - 2(T) - 0(P)  
Credits: 4

Course Objectives:

This course introduces the students to various models of computation. The course deals with automata theory, computability theory and the basics of computational complexity theory.

Module – I

Introduction to the theory of computation.

Finite state automata – description of finite automata, designing finite automata, NFA, finite automata with epsilon moves, equivalence of NFA and DFA, regular expressions, regular sets, Moore and Mealy machines.

Regular grammars, pumping lemma for regular languages, closure properties of regular sets and regular grammars,

Applications of finite automata, decision algorithms for regular sets, minimization of FSA.

Module – II

Context Free Grammar – Derivation trees, ambiguity, simplification of CFLs, normal forms of CFGs.

PDA – formal definition, examples of PDA, Deterministic PDA.

Pumping lemma for CFGs, closure properties of CFLs, decision algorithms for CFGs.

Module – III

Turing machines - Chomsky classification of languages, formal definition of Turing Machine, language acceptability by TM, examples of TM.

Variants of TMs – multitape TM, Non-deterministic TM, offline TMs, equivalence of single tape and multitape TMs.

Module – IV

Recursive and recursively enumerable languages – properties recursive and r.e. languages.

Decidability - decidable and undecidable problems, Universal Turing Machine, halting problem, reducibility.

References:


**Internal Continuous Assessment** (*Maximum Marks* - 50)

- 50% - Tests (minimum 2)
- 30% - Assignments (minimum 2) such as class room/home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.
- 20% - Regularity in the class

**University Examination Pattern:**

Examination duration: 3 hours  
Maximum Total Marks: 100

The question paper shall consist of 2 parts.

**Part A (20 marks)** - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

**Part B (80 Marks)** - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

**Course Outcome:**

At the end of the course, the students will have a good understanding of how efficiently problems can be solved on various models of computation. They will also have an idea as to whether a given problem is solvable on a particular model of computation.
13.506 SOFTWARE ARCHITECTURE & DESIGN PATTERNS (F)

Teaching Scheme: 2(L) - 1(T) - 0(P)  
Credits: 3

Course Objective:

This course introduces the students to UML and reusable objects, and provides them with the ability to design and apply software patterns.

Module – I

Software architectural attributes – attribute types.
Software architectural styles - pipes & filters, layered, event-based, data-centered, interpreter, MVC, message dispatcher, multi-tier distributed.
Introduction to UML - software architectural notations in UML.

Module – II

User interface architecture and design.
Domain-specific architecture – concepts, domain engineering.
Evaluation of architectural design.

Module – III

Object-oriented design - object identifications - static connection - dynamic connection.
Pattern oriented design - creational patterns - structural patterns - behavioural patterns.

Module – IV

Component oriented design - interface of components - implementation of interface - assembly and deployment of components - connections between components.
Evaluation of Software Detail Design.

References:

1. Gamma E., R. Helm, R. Johnson and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 2007.

**Internal Continuous Assessment** *(Maximum Marks-50)*

- 50% - Tests (minimum 2)
- 30% - Assignments (minimum 2) such as class room/home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.
- 20% - Regularity in the class

**University Examination Pattern:**

*Examination duration: 3 hours*  
*Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

**Part A (20 marks)** - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

**Part B (80 Marks)** - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

**Course Outcome:**

At the end of the course, the students will have gained the ability to analyze software problems and apply architectural patterns.
13.507 DIGITAL CIRCUITS LAB (F)

Teaching Scheme: 0(L) - 0(T) - 4(P)  
Credits: 4

Course Objective:

This course intends to provide hands-on experience to students in implementing digital circuits.

List of Exercises:

1. Realization of digital gates
2. Realization of flipflops
3. Design and implementation of a counter
4. Design and implementation of a shift register
5. Multiplexer / Demultiplexer
6. Timer Circuits (using 555)
7. Experiments using the 8051 microcontroller

Internal Continuous Assessment (Maximum Marks-50)

40% - Test

40% - Class work and Record (Up-to-date lab work, problem solving capability, keeping track of rough record and fair record, term projects, assignment, software/hardware exercises, etc.)

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours  
Maximum Total Marks: 100

Questions based on the list of exercises prescribed.

Marks should be awarded as follows:

20% - Algorithm/Design
20% - Implementing / Conducting the work assigned
30% - Output/Results and inference
30% - Viva voce

Candidate shall submit the certified fair record for endorsement by the external examiner.

Course Outcome:

At the end of the course, the students would have acquired the necessary hands-on skills to implement basic digital circuits.
13.508 DATABASE LAB (F)

Teaching Scheme: 0(L) - 0(T) - 4(P)  
Credits: 4

Course Objective:

This course intends to provide hands-on experience to students in data base management concepts.

List of Exercises: Programming exercises based on the courses 13.405 Data Base Design.

1. Familiarization of creation of databases and SQL commands (DDL, DML and DCL).
   Suitable exercises to practice SQL commands may be given.
2. Write a SQL procedure for an application which uses exception handling.
3. Write a SQL procedure for an application with cursors.
4. Write a DBMS program to prepare reports for an application using functions.
5. Write a SQL block containing triggers and stored procedures.
6. Develop a menu driven, GUI-based database application in any one of the domains such as Banking, Billing, Library management, Payroll, Insurance, Inventory, Healthcare etc. integrating all the features specified in the above exercises.

Internal Continuous Assessment (Maximum Marks-50)

40% - Test

40% - Regular lab work and proper maintenance of lab records

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours  
Maximum Total Marks: 100

Marks should be awarded as follows:

20% - Algorithm/ Design
20% - Programming/Implementation
30% - Output/Results and inference
30% - Viva voce

Candidate shall submit the certified fair record for endorsement by the external examiner.

Course Outcome:

At the end of the course, the students would have acquired the necessary hands-on skills to work on database management systems.