Rajagiri Certificate Program (RCP)

Rajagiri International School for Education & Research (RISER)

Course Details

Sister Institution of
Rajagiri School of Engineering & Technology

Index

1. RISER Certificate Program – CSIT Semester Plan

Semester 1

Course Title	Remarks	Credits
Introduction to Information Technology	Core	12
Introduction to Programming	Core	12
Software Engineering Fundamentals	Student Elective	12
Engineering Mathematics 1	Student Elective / IT Elective	12

Semester 2

Course Title	Remarks	Credits
Object Oriented Programming – Basics	Core	12
Database Concepts	Student Elective	12
Computer Organization	IT Elective	12
Data structures and Algorithms	Student Elective/ IT Elective	12

Semester 3

Course Title	Remarks	Credits
Advanced Java Programming	Core	12
Windows System Administration	IT Elective	12
Web Programming	Core	12
Usability and Professional Communication	Student Elective	12

Semester 4

Course Title	Remarks	Credits
Database Administration	Student Elective / Adv IT Elective	12
E Commerce	Advanced IT Elective	12
Computer Networks	IT Elective	12
Project Management	Student Elective	12

2. RCP General Guide Lines

2.1 Attendance

A minimum attendance of 75% is compulsory for all the courses. (Separate for the lecture and lab session) Students who have attendance below this limit will not be eligible to sit for the exam.

2.2 Grading and marks

Exams for all the courses will be in 100 marks (or it will be converted to 100). The mark grade conversion is as follows.

HD - (80 - 100) High Distinction
DI - (70 - 79) Distinction
CR - (60 - 69) Credit
PA - (50 - 59) Pass
NN - (00 - 49) Fail
RW - Result Withheld (see lecturer for reasons)
DNS - Did Not Sit/Did Not Submit

2.3 Teaching Methods:

The course will be based on the following teaching and learning activities:

• Lectures covering the theoretical part using Power Point presentations, Marking Pen Board, Smart Board, Black Board etc.

• Review questions at the beginning of the lecture and at the end of the lecture (in between in needed)

• Case studies

• Lab sessions

2.4 Evaluation Plan:

Students will be evaluated in this course using a combination of assessment methods, including:

• Written exams:
• Final Exam
• Lab Exams
• Single / Group Assignments
• Single / Group projects

It varies for each course. You will get the details from the faculty.

2.5 Applications for extensions

Applications for extensions for submission of assignment work must be made in writing (electronic mail is acceptable) to the lecturer no less than 24 hours before an assignment is due. Unless reasonable extenuating circumstances are claimed, and backed up by documentary evidence, extensions are not granted without penalty.

2.6 Teaching Resources:

Main Textbook

- Will be prescribed by the faculty

Supplementary Textbooks

- Will be prescribed by the faculty

Reading Assignments:

- Can use the reference books in the library.
- Handouts will be provided by the faculty;

Course Materials

- Will be available in the intranet

2.7 Classroom policies

Class participation and discussions are strongly encouraged. However, please be considerate to others: avoid coming to class late, leaving early, talking to other students, etc. Please turn off your laptops before the class starts. (If not mentioned to keep it on)

2.8 Overview of Learning Resources

You will make extensive use of computer laboratories and relevant software provided by the School. You will be able to access course information and learning materials through the intranet and may be provided with copies of additional materials in class or via email. Lists of relevant reference texts, resources in the library and freely accessible Internet sites will be provided. Students can borrow text books from the RCP library which is in the main building

3. Course Description

3.1 Introduction to Information Technology

Course Description

This course introduces students to the fundamentals of computer hardware and architecture, system software, application software, common computer networking systems and the internet. Students come to be aware of the component design of a computer, recognize common types of computer system and application software, identify various types of common computer network systems and understand the theory behind the world's biggest computer network, commonly known as the internet.

Learning Outcomes

On completion of this course you should be able to:

- Familiar with different operating systems – Unix, Linux
- Create Web page with personal details
- Describe the different components of a computer network
- Prepare a professional and comprehensive written report
- Demonstrate competence in the use of project management software.

3.2 Introduction to Programming

Course Description

Introduction to Programming teaches students to program using the C programming language. As an introduction, this course gives students an overview of the different components of the computer, different number systems and its conversions and Problem-solving strategies. Lecture materials are reinforced with hands-on lab exercises.

Learning Outcomes

On completion of this course you should be able to:

- Read and analyse problem descriptions.
- Analyse the input and output needs for a specified problem.
- Develop or choose appropriate algorithms for solving problems.
- Design algorithms using the control structures of structured programming.
- Describe the relationship between algorithm design and computer programming in modern high-level programming languages.

3.3 Software Engineering Fundamentals

Course Description

The course introduces the fundamentals of software engineering with an emphasis on software life cycle models, analysis and structuring of software development problems, project management techniques and contemporary design notations. It also introduces the fundamentals and methods used in managing software development. Topics will include software project planning and control, subcontract management, configuration management, risk management and organizing and managing software teams and commitment management. Tools used in analysis and design of software systems will be discussed within the context of a case study.

Learning Outcomes

On completion of this course you should be able to:

- Understand history, ethics and risks of software engineering,
- Understand principles of software process management,
- Understand principles of software process tasks management, and software life cycle.
- Understand the process of analysis and design using the object-oriented approach.
- Understand how to evaluate requirements for a software system.
- Be aware of current trends in the area of software engineering.

3.4 Object Oriented Programming – Basics

Course Description

This course teaches the fundamental ideas behind the object-oriented approach to programming; through the widely-used Java programming language. Concentrating on aspects of Java that best demonstrate object-oriented principles and good practice, you'll gain a solid basis for further study of the Java language, and of object-oriented software development. As well as providing grounding in the use of Java, the course will cover general principles of programming in imperative and object oriented frameworks.

Learning Outcomes

On completion of this course you should be able to:

• Describe the difference between applets and applications
• Look at Java source code and identify the variables, methods, and constructors
• Describe the difference between a class and an instance
• Describe the implications of inheritance and overloading
• Describe the implications of the static, final, const, public and private modifiers
• Develop simple algorithms and implement them using the standard control structures
• Develop simple algorithms and implement them using the standard control structures

3.5 Database Concepts

Course Description

Concepts, database architecture, relational algebra, the relational model and the normalization process, functional dependencies, database integrity and security, concurrent operations on database, distributed database systems architecture, object-oriented database approach, and deductive databases, relational query evaluation, XML databases.

Learning Outcomes:

- Gained a good understanding of fundamentals of database modeling and design, SQL query language, and implementation of a simple database application.

- Implement the structural constraints of relationships and how to perform them practically.

- Distinguish between the different types of attributes, primary keys, and foreign keys, super keys … etc.

- Gain the skill of drawing the ER-Diagrams, EER-Diagrams, some concepts of UML Diagrams as well.

- Apply transaction processing and concurrency in multi-user database systems to obtain accurate results.

3.6 Computer Organization

Course Description

This course provides basic and generic knowledge of digital computer organization and design at the machine and microprogramming levels with the associated assembly language programming concepts. It highlights the organization and architecture of the central processing unit, input-output peripherals, memory unit, and pipelining and parallel processing. It also include interfacing and communication of computer elements, functional organization, multiprocessor and alternate architectures, and performance enhancements.

Learning Outcomes:

On successfully completing this course, students will be able to:

- Understand the basic organization of modern computer systems.

- Understand how computer programs are organized, stored, and executed at the machine level.

- Understand the operation of fixed- and floating-point arithmetic units.

- Analyze an instruction-set architecture and propose a suitable datapath and control unit implementation.

- Understand how instruction pipelining enhances processor performance.

- Understand the basic organization of the memory hierarchy.

- Understand the input/output mechanisms used to connect computers to their external environments.
- Understand how system buses link the components of a computer system.

3.7 Data structures and Algorithms

Course Description

The purpose of this course is to introduce students to the topics of data structures and algorithm design along with their respective applications. The topics that will be covered are: arrays/linked lists, stacks, queues, recursion, trees, heaps, searching, sorting and hashing. The use of data structures in programming languages and relevant aspects of data and file management will be illustrated by using a high level language, which will be taught and used for the practical content of this course.

Learning Outcomes:

After successful completion of this course one should be able to

- Describe the basic data structures, algorithms and programming techniques, together with lists, stack, queues, search trees, hash tables, different sorting algorithms, search algorithms for graphs, and dynamic programming.
- Apply the techniques from the course when solving programming/algorithm problems. These techniques include methods for sorting and searching, basic graph algorithms, recursion, dynamic programming, and time and space analysis of programs, both basic techniques and amortized analysis.
- Select the best algorithm and/or data structure when solving a given programming problem.
- Analyze time and space required for the execution of a program, as well as the correctness of a program.
- Formulate a given programming task as an algorithmic problem, in order to select the best method for solving it.
- Combine and modify algorithms and data structures, in order to propose an efficient program**.**

3.8 Advanced Java Programming

Course Description

This course presents the basic concepts of object-oriented programming using Java. The course features an "objects first" approach to object-oriented programming (OOP), starting with objects, classes and methods, inheritance, interfaces and polymorphism.

Learning Outcomes:

After successful completion of this course one should be able to

On completion of this course you should:

• use standard Java classes and interfaces;
• use object oriented program development framework ;
• develop simple algorithms and implement them using the standard control structures;
• use arrays and other container classes for storing and manipulating objects;
• write programs that promote code reuse;
• refine the design using step wise refinement
• correctly manipulate standard data files, focusing on text files;
• handle exceptions thrown and write exception classes;
• follow good coding guidelines
• devise strategies to test the software developed.

3.9 Windows System Administration

Course Description

This is an introduction to the process of choosing, installing, configuring and maintaining Microsoft Windows client and server systems. Topics include user management, file systems, network domains and domain management, mailers, and printing. Students get practice in writing scripts for performing maintenance tasks. Also, students learn how these tasks fit into the more general system administration process.

Learning Outcomes

On completion of this course you should be able to:

- Install and configure both the windows server and windows workstation operating systems.
- Configure server side services including: windows active directory, file sharing/security, exchange email services, remote desktop, WSUS server.
- Configure workstation settings including: managing profiles, distribute applications and manage group policies.
- Window Network Administration.

3.10 Web Programming

Course Description

Introduces fundamental programming techniques using current web development software. Students design procedures and write computer instructions to solve business problems, learn procedural programming, develop graphical user interfaces for the web and work with events and objects. The course will introduce you to Web protocols, HTML and XML, client side processing JavaScript and server side processing with PHP.

Learning Outcomes

On completion of this course you should have thorough understanding of:

• Java script programming language and DHTML.
• Event-driven programming.
• Form validation.
• Web browser objects, Java script objects and DOM objects.
• Design and implementation of websites using Java script and DHTML.
• Fundamental syntax of the programming language used in the course.
• Concepts of data types and dynamic data types.
• Dynamic arrays of objects.
• Concepts of assignment; arithmetic, relational, and Boolean expressions.
• Parameter passing and its implications.
• Debugging techniques, including the use of a symbolic debugger.
• Explain the functions of clients and servers on the Web.
• Design and implement an interactive web site with regard to issues of usability, accessibility and internationalization.
• Demonstrate the ability to retrieve data from a database and present it in a web page

3.11 Usability and Professional Communication

Course Description

This course introduces principles and strategies for effective communication in both academic and workplace engineering settings. Through analysis of case studies and of academic and professional genres, this course develops the oral and written communication skills that characterize successful engineers. Students will prepare professional documents that focus specifically on communicating academic and technical knowledge to diverse audiences.

Learning Outcomes

Upon completion of this course, it is expected that the students will..

- Produce written texts in a variety of professional genres - texts that communicate effectively and adhere to professional ethical standards.
- Deliver clear and professional oral presentations on a range of engineering topics.
- Reflect on and justify the rhetorical choices involved in planning, writing, revising, and presenting academic and professional engineering documents.
- Demonstrate the ability to work as part of a research team, coordinating workflow and collaboratively presenting outcomes.
- Synthesize the studious research and professional best practices related to an engineering project in the student's field.
- Produce and refine an array of personal professional documents.
- Demonstrate the capacity for life-long education through sustained reflection, revision, and research.

3.12 Database Administration

Course Description

Database-management systems, including database architecture, design, administration, and implementation. Evaluation and use of database-management systems for computers, with emphasis on microcomputer-based systems. Topics include DBMS installation and configuration issues; performance optimizing, monitoring and tuning; data movement and load balancing; database security, back-up and recovery; multi-platform DBA issues; and DBA tools.

Learning Outcomes

On completion of this course you should be able to:

- Analyze the DBMS requirements for specific scenarios
- Install, configure, optimize and tune the performance of a DBMS;
- Design and implement security, back-up and recovery measures

3.13 E Commerce

Course Description

This course covers tools, skills, business concepts, and social issues that surround the emergence of electronic commerce. The student will develop an understanding of the current practices and opportunities in EDI, electronic publishing, electronic shopping, electronic distribution, electronic collaboration and database issues. Other issues include standards, security, authentication, privacy, intellectual property, acceptable use, legal liability, and economic analysis.

Learning Outcomes

Upon successful completion of the course, each student will be able to:

- Define and discuss major concepts, tools, techniques, and methods of electronic commerce, with a focus on the web.
- Explain the difference between business to consumer (B2C) and business to business (B2B) models and be able to recognise and apply each model in a typical situation
- Plan, design, and prototype a significant e-commerce web project.
- Effectively use state of the art technologies (e.g. Java), and be able to combine them to develop a basic enterprise application

3.14 Computer Networks

Course Description

This course examines principles, design, implementation, and performance of computer networks. A focus will be placed on wireless networking, reflecting rapid advances in this area. Topics include: Internet protocols and routing, local area networks, wireless communications andnetworking, performance analysis, congestion control, TCP, network address translation, multimedia over IP, switching and routing, mobile IP, peer-to-peer networking, network security, Data communications, network architectures, communication protocols, data link control, medium access control.

Learning Outcomes

On successful completion of this course, you should be able to explain:

- Basic concepts of data communication;
- Network transport architectures, TCP, UDP, TCP congestion control
- Routing and forwarding, intra-domain and inter-domain routing algorithms
- Link layers and local area networks, Ethernet, WiFi, and mobility
- Network measurement, inference, and management
- Network security
- Network experimentation and performance analysis
- Protocol verification

3.15 Project Management

Course Description

Introduces the fair, theoretical, and practical challenges of the project management framework, including the basic project management phases (initiation, planning, execution, control, and closure) and interactions. Integrates project management and strategic management. Instruct students on the purpose and preparation of their portfolios.

Learning Outcomes

At the conclusion of this course you will have learnt about

- Project Initiation.
- Project Planning.
- Project Execution.
- Project Control.
- Project Closing.
- Technical risk of management
- Tools used to implement project management