College of Computer Science and Information Systems | **Department of Computer Science**

Course Code : 330CSS-3 | **Programming Paradigms**

Contact Hour : 3(0) | **Prerequisite : 113CSS-4**

Coordinator -

2. Course Description

Introduction to programming languages, the static and dynamic scope, communication between subprograms via parameter passing, and storage management (static and dynamic), languages using virtual machines: Java programming language is discussed as an example of languages that use virtual machines (VM); the main differences between C++ and Java, Introduction to functional programming (basic skills about Scheme programming language), general differences between the Scheme and the C programming language, logic programming, modern programming (e. g. Python and C#), Comparison among different Object Oriented Programming Languages.

3. Course Learning Outcomes

SL	By the end of this course, students should be able to:	Linkages to POs
1.	Describe the basics of functional programming, logic programming paradigms with proper examples	a(S),i(S)
2.	Discuss the scope and memory management concepts of various programming languages	i(S)
3.	Distinguish among different types of programming language paradigms	a(S)
4.	Analyze the syntactical differences of commonly used programming languages	b(S)
5.	Integrate main concepts of object oriented programming	c(S),i(W),j(S)
6.	Propose appropriate solutions for real-life problems with specific programming language	b(S),i(S)

4. Learning Resources

Text	Robert W. Sebesta, Concept of Programming Languages, Pearson Education, 10th Edition, 2012
Reference	Saroj Kaushik, Logic and Prolog Programming, New Age International.
Reference	Mark Lutz and David Ascher, Learning Python, Oâ€™REILLY and Associates, Latest Edition.
Reference	Anders Hejlsberg, Mads Torgersen, Scott Wiltamuth and Peter Golde, The C# Programming Language, Microsoft .NET Development Series, Latest Edition.
Reference	Joshua Bloch, Effective Java: Programming Language Guide.

5. Course Content : The list below provides a summary of the material that will be covered during the course

Week	Topics	References Book / Others Source	Special Event	Tutorial Activities	Lab Activities
1.	Introduction to Programming Languages : Reason to Study Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-offs, Implementation methods, Programmi	Textbook â€“ Chapter 1			Introduction to C++ programming

2.	Language Categories, Language Design Trade-offs, Implementation methods, Programming environment	Textbook â€“ Chapter 1			Introduction to C++ programming
3.	Data types â€“ Introduction, Primitive Data Types, Character String Types, Type Checking	Textbook-Chapter 6		Tutorial 1	Lab Activity -1 â€“ Writing simple programs in C++, Run program
4.	Static and Dynamic Scope â€“ Introduction, Names, Variables, Concept of Binding Storage Management (Static & Dynamic) â€“ Scope, Scope and Lifetime, Referencing Environments, Named Constants, Comparative Study	Textbook â€“ Chapter 5	Assignment 1 (5th week)	Tutorial 2	Lab Activity â€“ 2 & 3 Simple Call Returns, Subprograms, Recurs
5.	Static and Dynamic Scope â€“ Introduction, Names, Variables, Concept of Binding Storage Management (Static & Dynamic) â€“ Scope, Scope and Lifetime, Referencing Environments, Named Constants, Comparative Study	Textbook â€“ Chapter 5			Lab Activity â€“ 2 & 3 Simple Call Returns, Subprograms, Recurs
6.	Languages used as Virtual Machines (Java) â€“ Object Oriented Paradigm, Design Issues of Object Oriented Languages, Support for OOP in Java., Abstract Data Types and Encapsulation Constructs, Comparative Study	Textbook â€“ Chapter 11, 12	Midterm Exam-I (6th week)	Tutorial 3	Lab Activity 4 & 5 Static & Dynamic Scope, Communication betwee
7.	Languages used as Virtual Machines (Java) â€“ Object Oriented Paradigm, Design Issues of Object Oriented Languages, Support for OOP in Java., Abstract Data Types and Encapsulation Constructs, Comparative Study	Textbook â€“ Chapter 11, 12			Lab Activity 4 & 5 Static & Dynamic Scope, Communication betwee
8.	Introduction to Functional Programming Languages â€“ Fundamentals of Functional Programing Languages, LISP, An Introduction to Scheme, LISP, ML, Haskell, Support for Functional Programming in Primarily Imperative Languages, A Comparison of functional and i	Textbook â€“ Chapter 15	Lab Quiz (8th week)	Tutorial 4	Lab Activity 6 Function Overloading in C++. Converison of C++
9.	Introduction to Functional Programming Languages â€“ Fundamentals of Functional Programing Languages, LISP, An Introduction to Scheme, LISP, ML, Haskell, Support for Functional Programming in Primarily Imperative Languages, A Comparison of functional and i	Textbook â€“ Chapter 15			Lab Activity 6 Function Overloading in C++. Converison of C++

10.	Logic Programming â€“ Introduction, Overview of Logic Programming, Prolog â€“ Origins, Basic Elements, Deficiencies, Applicaton of Logic Programming, Comparative Study	Textbook â€“ Chapter 16		Tutorial 5	Lab Activity 8 & 9 Implementation of Polymorphism using virtual
11.	Logic Programming â€“ Introduction, Overview of Logic Programming, Prolog â€“ Origins, Basic Elements, Deficiencies, Applicaton of Logic Programming, Comparative Study	Textbook â€“ Chapter 16			Lab Activity 8 & 9 Implementation of Polymorphism using virtual
12.	Modern Programming â€“ C#, Comparative Study	To be mentioned		Tutorial 6	Lab Activity 10 Introduction to basic programming in C#
13.	Modern Programming â€“ C#, Comparative Study	To be mentioned			Lab Activity 10 Introduction to basic programming in C#
14.	Revision		Final Lab Exam		

6. Evaluation Scheme: The following list is the contribution of course components to the final grade for the course.

Component	Weight (%)
Assignments	5%
Quizes	5%
Lab Test & Performance	10%
First Midterm Exam	15%
Second Midterm Exam	15%
Lab Final Exam	10%
Final Examination	40%
Total	100