

# **Faculty of Computers and Information Technology**

# **Fundamentals of Programming Languages**

Information :

Course Code :	CS314	Level	:	Undergraduate	Course Hours :	3.00- Hours

**Department :** Department of Computer Science

#### Instructor Information :

Title	Name	Office hours
Associate Professor	Osama Fathy Saleh Hegazy	5
Teaching Assistant	Rahmatallah Hossam Farouk Hassan Mohamed AlSofany	2

#### Area Of Study :

Apply the main concepts of programming languages.

Combine and evaluate different types of programming languages with a practical support for these features. Analyze the requirements of language design trade-off and design a solution for these requirements.

Compare and evaluate methodologies from range of major programming languages.

Use effectively communication skills.

Understand knowledge that enhances skills in describing syntax, grammars, static and dynamic semantics. Develop and evaluate a computer based system process and components.

#### **Description :**

Describing syntax and semantics. Identifiers: names, binding, type checking, and scopes. Data types, subprograms and their implementation, concurrency, programming paradigms such as declarative programming, object oriented programming and component programming, parallel and distributed programming. Comparison and evaluation of different programming languages.

### Course outcomes :

a.Knowled	ge and Understanding: :		
1 -	Discuss the fundamental concepts of programming languages and their different types such as procedural, object oriented, declarative, service-oriented programming languages.		
2 -	Identify the criteria for current use of BNF and parse trees for languages parsing.		
3 -	Explain functional requirements and constrains for memory usage that applied for different types of programming languages		
4 -	Define the fundamental variables related concepts like Binding, Lifetime, and Scope.		
b.Intellect	ual Skills: :		
1 -	Analyze programs execution according to approaches used for parameter passing, and variables scoping and lifetimes.		
2 -	Select appropriate methodologies and techniques to plan and assess when to use each programming paradigm.		
0	Enders and the fifth of the state of the sta		



## c.Professional and Practical Skills: :

1 -	Apply the different design methodologies using declarative and object-oriented programming language.
2 -	Analyze, design, implement, test, maintain and manage stack, scoping, life time assessment, and parameter passing
3 -	Apply BNF for a given programming language

# d.General and Transferable Skills: :

1 -	Work in a team effectively and efficiently considering time.
-----	--

2 - Apply communication skills and techniques in presentations and report writing for range of audiences using various methods and tools.

## **ABET Course outcomes :**

1 -	Understand the main concepts of programming languages.
2 -	Compare and evaluate different types of programming languages.
3 -	Analyze the requirements of language design trade-off and design a solution for these requirements.
4 -	Describe syntax, grammars, static and dynamic semantics.
5 -	Develop and evaluate a computer-based system.

# **Course Topic And Contents :**

Different types of programming languages422Main Concepts, reasons for learning concepts of programming languages, programming domains languages evaluations criteria.422Language design trade-offs and implementation methods.4222Syntax and semantics, formal methods of describing syntax, extended BNF, attribute grammars, static and dynamic semantics.422Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.222Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.222Evaluation of major programming languages (p1)4222Mid Term Exam2222Evaluation of major programming languages (p2)4222Object Oriented Programming languages (p2)4222Object Oriented Programming languages (p2)4222Object Oriented Programming: Object Oriented Programming- Design Issues-examples222Final Exam2222Final Exam2222Final Exam2222Final Exam2222Final Exam2222Final Exam2222Final Exam2222Final Exam2222 <td< th=""><th>Торіс</th><th>No. of hours</th><th>Lecture</th><th>Tutorial / Practical</th></td<>	Торіс	No. of hours	Lecture	Tutorial / Practical
Main Concepts, reasons for learning concepts of programming languages, programming domains languages evaluations criteria.422Language design trade-offs and implementation methods.422Syntax and semantics, formal methods of describing syntax, extended BNF, attribute grammars, static and dynamic semantics.422Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.222Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.422Evaluation of major programming languages (p1)4222Statement-Level Control Structure: Selection, iterative, unconditional branching.222Mid Term Exam222Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Besign Issues-examples222Final Exam222	Different types of programming languages	4	2	2
Language design trade-offs and implementation methods.422Syntax and semantics, formal methods of describing syntax, extended BNF, attribute grammars, static and dynamic semantics.422Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.222Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.422Evaluation of major programming languages (p1)422Statement-Level Control Structure: Selection, iterative, unconditional branching.222Mid Term Exam222Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Besign Issues-examples222Revision4222Final Exam2222	Main Concepts, reasons for learning concepts of programming languages, programming domains languages evaluations criteria.	4	2	2
Syntax and semantics, formal methods of describing syntax, extended BNF, attribute grammars, static and dynamic semantics.422Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.422Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.422Evaluation of major programming languages (p1)422Statement-Level Control Structure: Selection, iterative, unconditional branching.422Mid Term Exam222Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples222Revision4222Final Exam2222Statemant2222Final Exam2222Statemant2222Statement-Level Control Structure: Selection, iterative, unconditional branching.422Statement-Level Control Structure: Selection, iterative, unconditional branching.222Statement-Level Control Structure: Selection, iterative, unconditional branching.222Statement-Level Control Structure: Selection, iterative, unconditional branching.222Statement2222Statement22222St	Language design trade-offs and implementation methods.	4	2	2
Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.422Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.422Evaluation of major programming languages (p1)4222Statement-Level Control Structure: Selection, iterative, unconditional branching.4222Mid Term Exam2222Evaluation of major programming languages (p2)4222Object Oriented Programming: Object Oriented Programming- Design Issues-examples4222Revision42222Final Exam22222Statemant22222Diget Oriented Programming: Object Oriented Programming- Design Issues-examples4222Revision42222Final Exam22222Statement42222Statement Statement42222Statement Statement42222Statement Statement42222Statement Statement42222Statement Statement42222Statement Statement42222Statement Sta	Syntax and semantics, formal methods of describing syntax, extended BNF, attribute grammars, static and dynamic semantics.	4	2	2
Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment422Evaluation of major programming languages (p1)422Statement-Level Control Structure: Selection, iterative, unconditional branching.422Mid Term Exam211Exceptional handling and event handling Levaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples222Revision4222Final Exam2222	Bindings and Scopes: Names, variables, the concept of binding, scope and lifetime.	4	2	2
Evaluation of major programming languages (p1)422Statement-Level Control Structure: Selection, iterative, unconditional branching.422Mid Term Exam222Exceptional handling and event handling422Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples22Revision422Final Exam22	Data Types: character strings, arrays, records, pointers, heap, expressions, overloaded operators, type conversion, assignment statements.	4	2	2
Statement-Level Control Structure: Selection, iterative, unconditional branching.422Mid Term Exam222Exceptional handling and event handling422Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples22Revision422Final Exam222	Evaluation of major programming languages (p1)	4	2	2
Mid Term Exam2Image: Constraint of the state of t	Statement-Level Control Structure: Selection, iterative, unconditional branching.	4	2	2
Exceptional handling and event handling422Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples222Revision4222Final Exam2222	Mid Term Exam	2		
Evaluation of major programming languages (p2)422Object Oriented Programming: Object Oriented Programming- Design Issues-examples422Revision4222Final Exam22422	Exceptional handling and event handling	4	2	2
Object Oriented Programming: Object Oriented Programming- Design Issues-examples422Revision422Final Exam221	Evaluation of major programming languages (p2)	4	2	2
Revision422Final Exam22	Object Oriented Programming: Object Oriented Programming- Design Issues-examples	4	2	2
Final Exam 2	Revision	4	2	2
	Final Exam	2		

# **Teaching And Learning Methodologies :**

Interactive Lectures including Discussions

Practical Lab Sessions



## Self-Study (Project / Reading Materials / Online Material / Presentations)

#### **Case Studies**

**Problem Solving** 

Course Assessment :				
Methods of assessment	Relative weight %	Week No	Assess What	
Assignments	5.00	4		
Final Exam	40.00	14		
Midterm Exam (s)	30.00	9		
Practical Exam	10.00			
Presentations	5.00	12		
Quizzes	10.00	5		
Team Work Projects	5.00	12		

# Course Notes :

An Electronic form of the Course Notes and all the slides of the Lectures is available on the Students Learning Management System (Moodle)

#### Recommended books :

P. Roy, S. Haridi, Concepts, Techniques and Models of Computer Programming, 1st ed. The MIT Press, 2004. ISBN: 978-0262220699.

# Web Sites :

http://www.mindview.net/Books/TIJ/DownloadSites