CS1901775 ***Programming Languages Design***

Instructor: Dr. Abdel latif Abu Dalhoum
E. mail: a.latif@ju.edu.jo

Course Description

Syntax and Semantics, Comparison and Design of Programming Languages, Structure of Compiled and Interpretive Languages, Data types and Abstract Data types, Control Structures, Language Features in Programming, Object-Oriented Programming, Syntax and Translation Semantics and properties for real and abstract Machines and Formal Semantics.

Course Objectives

A- Knowledge and understanding

A1- Understand the nature of natural and programming languages.
A2- Understand basic goals, principles and problems of language design and evaluation.
A3- Understand the formal description of programming languages. Learn the formal semantic description methods and how they are used in language design and implementation.
A4- Understand the different categories and paradigms (programming styles) of high level languages.
A5- Understand the programming languages constructs (elements) and their semantics.

B- Intellectual Skills.

B1- Recognize design goals and evaluation criteria of programming languages
B2- Analyze and compare between different formal approaches to describe the semantics of high level languages
B3- Analyze and compare between the implementation of same features in different programming languages. Emphasis on type systems and generics.
B4- Recognize strengths and weaknesses of the individual constructs of programming languages.
B5- Analyze the current research and trends in language design, description and implementation.

C- Subject specific skills

C1- Evaluate programming languages with regard to their features, design and implementation.
C2- Design, formally describe and implement various aspects of programming languages.
C3- Implement execution models of different language paradigms.
C4- Implement syntax and semantic extension of a programming language.
C5- Apply the learned techniques in other areas of computer science.

D- Transferable skills

D1- Discuss and work in a group in order to explore the current research and development of programming languages, including their formal semantics, design and implementation.
D2- Discuss and work in a group in order to design a new language and extend existing ones.
D3- Document and present the above mentioned work.

Contents:

1. The Nature of Language

1.1 Communication
1.2 Syntax and semantics
1.3 Natural languages and programming languages.
1.4 The standardization process.
1.5 Nonstandard compiler

2. Representation and Abstraction

2.1 What is a program?
2.2 Representation.
2.3 Language design
2.4 Classifying languages

3. Elements of Language

3.1 The parts of speech
3.2 The met language

4. Formal Description of Language

4.1 Foundations of programming languages
4.2 Syntax
4.3 Semantics
4.4 Extending the semantics of language

5. Primitive Types

5.1 Primitive hardware types
5.2 Types in programming languages
5.3 A brief history of type declarations

6. Modeling Objects

6.1 Kinds of objects
6.2 Placing a value in a storage object
6.3 The storage model: managing storage objects

7. Names and Binding

7.1 The problem with names
7.2 Binding a name to a constant
7.3 Survey of allocation and binding
7.4 The scope of a name
7.5 Implications for the compiler/interpreter

8. Expressions and Evaluation

16. Modeling Objects

16.1 The purpose of modules
16.2 Modularity through files and linking
16.3 Packages in ada
16.4 Object classes

17. Modeling Objects

17.1 Generics
17.2 Limited generic behavior
17.3 Parameterized generic domains

18. Dispatching with Inheritance

18.1 Representing domain relationships
18.2 Sub domains and class hierarchies
18.3 Polymorphic domain and functions
18.4 Can we do more with generics?

Teaching methods

Method	Lecture	Demo	Laboratory	Case study
Learning objective	A1-A5 +B1-B4	B2+B5+D3	C2+C3+C4+C5	C1-C2+D1-D2
Assessment	Exams+assignments	Exams+assignments	Presentation	Project+Presentation

Evaluation:

Projects and Software Assignment: ***30***%

Individual and group projects in the following:

- Application of the above concept in major programming languages.
- Current-research trends in the design and implementation of high level languages.
- Advances in language design, formal semantics, implementations and development.
- Implementation of execution model and extension of assigned languages.

Midterm Exam: 30%
Final Exam: 40%

Text Book and References

1. Alice E. Fisher and Frances S. Grodzinsey, The Anatomy of Programming
2. Terrence W. Pratt and Marvin V. Zelkowitz, Programming Languages Design and Implementation Prentice Hall, 2001.
3. Carlo Ghezziqnd Mehdi Jazayeri, Programming Language Concepts, Willey, 1998.
4. Bruce J. Macleuhour, Principles of Programming Languages: Design, Evaluation and Implementation, Oxford University Press, 1999.
5. Michael L. Scott, Programming Language Pragmatics, Morgan Kaufman Publishers, 2000.
6. References on Selected Programming Languages: Java, C^{++}, Smalltalk, ML, Lisp, Prolog.
7. References on selected formal semantics description methods.
8. Selected research papers and publications covering programming language design, implementation and semantics.

CS1901775 Subjects

1. The Nature of Language
2. Representation and Abstraction
3. Elements of Language
4. Formal Description of Language
5. Primitive Types
6. Modeling Objects
7. Names and Binding
8. The Representation of Types
9. The Semantics of Types
10. Modeling Objects
11. Generics