CSC422 - Concept of programming Languages [No26]

Robert W. Seebesta

1. why study the course
2. Genealogy of some selected programming languages (Fortran, Java, C#, Python, Ada, PHP)
3. Syntax Representation(BNF/EBNF)
4. Expression
5. Statements
6. Object Orientation
7. Exception and Event Handling

Why study This course

The True essence of this course is to acquaint us with the various programming languages, features and statements such that we see that semantics and pragmatics of programming languages will be sufficiently exposed to us to enable us:

1. Make you a better programmer:
A programmer that knows and writes program in a particular language. If he comes across a feature in another programming language and the programming language he knows does not have that feature, then He'll like to find a way to implement it in his my own language.

2. Make informed choices of programming languages for a given software program.

3. Will make you learn new programming languages easily:
This will make us know what to watch out for when learning a new programming language.

4. It will help us design new and better programming languages
5. Will give a good understanding of implementation issues
6. Will lead to overall advancement of computing

Note: Syntax, Semantics, Pragmatics

Programming language Evaluation Criteria

Basically we can evaluate programming languages using the following criteria:

1. Readability: Its the ease with which an average programmer with a given programming language can comprehend the codes written in that programming language

2. Writeability: Its the ease with which an average programmer can computationally express himself/

herself in that programming language.

3. Reliability: The difficulty with which an amateur programmer can introduce error into the code while programming is called reliability..

Error: while error is as a result of human intellectual and computational deficiencies
Exception: is as a result of natural or environmental negligence necessarily outside the control of a programmer.

Attributes of Error and Exception

Exception
1. exception must occur
2. can't be corrected but handled
3. natural and environmental negligence(exception)
4. Exceptions are not avoidable

Error
1. Error will not occur
2. Error can be corrected
3. Due to human intellectual and computational deficiencies
4. Error are avoidable

4. Cost: (Implementation, learning, compilation , community support)

Every programming language must have the following characteristics:
1. Uniformity: A language characteristic that guarantees consistency in patterns and rules underlining the programming language.
2. Simplicity: Has to do with the number of primitive features of the programming language. If the set of the primitive features is too large, then the language becomes to hard to learn (c++).

3. Orthogonality: The Degree to which lower level features of a programming language can be combined to form higher level features.

4. Generality: A programming language should be capable of being used to solve any tractable computational problem.

5. Support for abstraction: Abstraction makes an entity appear simple and is used as such.
If a programmer has written a complex procedure, the abstraction there is the use of the procedure's signature or name.. it makes the entire complex procedure look simple. especially during implementation.

6. Exception Handling Support: It gives support for handling exceptions in a programming language

7. Syntax design : Because learning a pprogramming language is dependent on the syntax, so we pay attention to how this syntax is designed..

- Syntax should be design to be consistent with our norm and natural way of reasoning

ASSIGNMENT

Genealogy of some selected programming languages

How the language started

The fault it had

What was added (at every stage)

Why it was added(at every stage)