**CSE4060 - Principles of Programming Languages**

**PRESENTATION PROJECT (Slides due 30.12.2019)**

You will make a presentation on a programming language from the given list of languages. Each group will present a different programming language. You should reserve the language you select from the following pool. First-come first-served basis will be used. You can work in groups of at most three.

The language pool:

* Javascript
* Python
* Ruby
* PHP
* Matlab
* Swift
* Rust
* Elixir
* Scala
* R
* Java
* C++
* Or a language you choose

You will introduce the programming language, including the following information at least:

* a brief history of the language
* types of applications/industries the language is used for
* the superiority of the language among other languages
* …

The presentation should also answer the following questions. It is preferred that you give examples to support each answer.

* Are names case sensitive?
* Are the special words of the language reserved words or keywords?
* Is using aliases allowed?
* How is a type specified?
* When does type binding take place?
  + If static, does it use explicit or implicit type declarations?
* If implicit type declaration is used, how are the types bound?
* Which category of variables are used? (Static, stack-dynamic, explicit heap-dynamic, implicit heap-dynamic)
* Where does the declaration statements appear?
* Does it use static or dynamic scoping?
* Which primitive data types are provided?
* If string is a primitive type, should strings have static or dynamic length?
* Does it include enumeration type?
* Is an enumeration constant allowed to appear in more than one type definition, and if so, how is the type of an occurrence of that constant checked?
* Are enumeration values coerced to integer?
* Any other type coerced to an enumeration type?
* What types are legal for array subscripts?
* Are subscripting expressions in element references range checked?
* When are subscript ranges bound?
* When does array allocation take place?
* Are ragged or rectangular multidimensional arrays allowed, or both?
* Can arrays be initialized when they have their storage allocated?
* What kind of slices are allowed, if any?
* Does it include associative arrays? How?
* How are the records used?
* Are elliptical references allowed?
* Does the language support tuples?
* Does the language support unions?
* Does the language support pointer types, reference types, or both?
  + What are the scope of and lifetime of a pointer variable?
* For arithmetic expressions:
  + What are the operator precedence rules?
  + What are the operator associativity rules?
  + What is the order of operand evaluation?
  + Are there restrictions on operand evaluation side effects?
  + Does the language allow user-defined operator overloading?
  + What type mixing is allowed in expressions?
* Does the language use short-circuit evaluation?
* For two-way selection statements:
  + What is the form and type of the control expression?
  + How are the **then** and **else** clauses specified?
  + How should the meaning of nested selectors be specified?
* For multiple-selection statements:
  + What is the form and type of the control expression?
  + How are the selectable segments specified?
  + Is execution flow through the structure restricted to include just a single selectable segment?
  + How are case values specified?
  + How should unrepresented selector expression values be handled, if at all?
* For counter-controlled loops:
  + What are the type and scope of the loop variable?
  + Should it be legal for the loop variable or loop parameters to be changed in the loop body, and if so, does the change affect loop control?
  + Should the loop parameters be evaluated only once, or once for every iteration?
* For logically controlled loops:
  + Should the control be pretest or posttest?
  + Should the logically controlled loop be a special case of the counting loop statement or a separate statement?
* Does the language support used-located loop control mechanisms? How?
* Does the language support iteration based on data structures? How?
* Considering the subprograms:
  + Are local variables statically or dynamically allocated?
  + Can subprogram definitions appear in other subprogram definitions?
  + What parameter passing methods are provided?
  + Are the types of the actual parameters checked against the types of the formal parameters?
  + If subprograms can be passed as parameters and subprograms can be nested, what is the referencing environment of a passed subprogram?
  + Are functional side effects allowed?
  + What types of values can be returned from functions?
  + How many values can be returned from functions?
  + Can subprograms be overloaded?