

PPL-MID

Code Style

What are the major language evaluation criteria?

- Readability
- Writability
- Reliability
- Cost

Many languages distinguish between uppercase and lowercase letters in user-defined names. What are the pros and cons of this design decision?

- Pros: More variables. It is easy to understand what the variable means.
 - Cons: It makes programs less readable, because words that look very similar are actually completely different, such as **SUM** and **Sum**.
-

Declaration

How do type declaration statement for simple variables affect the readability of language, considering that some languages do not require them?

- Type declaration statement have very little effect on the readability of programs.
- If no type declaration, it may be an aid to readability.
- In a program, the declaration of a variable must be found before the reader can determine the type of that variable when it is used in the program.

The type of a variable can determine what properties of it?

- The range of values the variable can store
 - The set of operations that are defined for values of the type.
-

Language Design

Language design is majorly influenced by what factors?

- Computer Architecture.
- Programming Methodologies.

What is the function of a grammar?

A grammar describes the syntax of a programming language.

Java uses a right brace to mark to end of all compound statements. What are the arguments for and against this design?

- using the right brace `}` to close all compounds statements is "simplicity."
- the location of its matching left brace `{` is not always obvious (hard to debug).

What are "special words", "keywords" and "reserved words"?

- Special words in programming languages are used:
 - to make programs more readable by naming actions to be performed.
 - to separate the syntactic entities of programs.
- A keyword is a word of a programming language that is special only in certain contexts.
- A reserved word is a special word of a programming language that can NOT be used as a name.

If **INTEGER** and **REAL** are keywords in a language, then are the following statements correct?

```
INTEGER REAL;  
REAL INTEGER;
```

YES. (Teacher took **FORTRAN** as an example.)

If **INTEGER** and **REAL** are reserved work in a language, then are the following statements correct?

```
INTEGER REAL;  
REAL INTEGER;
```

NO.

What arguments can you make against the idea of a single language for all programming domains?

Since every language was designed for a particular goal. The language would necessarily be huge and complex; the language would probably not be very good for any programming domain; Compilers would be expensive and costly to maintain.

What is the lifetime of a variable?

The lifetime of a variable is the time during which the variable is bound to a specific memory location.

When does the lifetime of a variable begin and end?

- BEGIN: When variable is bound to a specific cell.
- END: When variable is unbound from memory cell.

What are the major drawbacks of implicit declaration?

Detrimental to reliability because they prevent the compilation process from detecting some typographical and programmer errors.

In an imperative language like C,

- Is a non-local variable name appearing in different functions behind(bound?) to the same physical memory cells?
 - No.????
 - Is a non-static local variable name of a function bound to the same physical memory cell in different execution of that function?
 - No.
-

Stack, Heap (Memory)

In order to provide recursive subprograms, what kind of variables must a programming language provide?

Stack-dynamic.

Address

(1) What is the L-value of a variable? (2) What is the R-value of a variable?

- (1) the address of the variable.
 - (2) The content of the variable.
-

Aliases

What are aliases?

It is possible to have multiple variables that have the same address.

What are the disadvantages of aliases?

- A hindrance to readability because it allows a variable to have its value changed by an assignment to a different variable.
 - Makes program verification more difficult.
-

Compiler Join

If the data types of all variables in a program written in a language can be determined statically, and a compiler for that language makes only static type checking, could the compiler detect all type errors in a program written by that language? Explain your answer.

No. If the language allows a variable to store values of different types, then dynamic type checking must be used to detect type errors. For instance, union in C.

What is the purpose of type checking?

The activity of ensuring that the operands of an operator are of COMPATIBLE types.

From the point of view of an operation, what is the definition of a compatible type?

Legal for the operator or allowed under language rules to be implicitly converted by compiler-generated code (or the interpreter) to a legal type.

What is the definition of type errors?

The application of an operator to an operand of an inappropriate type.

List all the major components of a compiler and their output?

Major components of a compiler	Output
lexical analyzer	lexical units
syntax analyzer	parse tree
intermediate code generator, semantic analyzer	intermediate code
code generator	machine code

Code

(a) What error does the following C program have? (b) Why does the program have the error?

```
#include <stdio.h>
void foo (unsigned int a) {
    unsigned int b;
    b = a - 123;
}
void main() {
    int c = 789;
    foo(c);
}
```

- (a). An **integer** variable is passed to an **unsigned integer** variable.
- (b). An integer value is transmitted to an unsigned integer. Hence, a negative value will be treated as a large positive value.

For the following c assignment statement, **grade = study + 20**; What is the binding time of the following attributes?

- The type of variable study. At compile time.
- The set of possible values of variable grade. At compiler design time.
- The internal representation of the literal 20. At compiler design time.
- The meaning of operator "+". At compile time.
- The value of variable grade. At execution (run) time.

Supposing **Ada** used strict name type equivalence, consider the following **Ada** code:

```

type Subtype is 100..300;
type Subtype is 100..300;
times : Subtype;
length : Subrange;

```

- (1) Could we assign the value of variable `times` to variable `length`?
 - NO.
- Give your explanation.
 - The types of these two variables are different.

For each variable or parameter in the above program, list the segment(e.g. data segment, stack segment, or BSS segment) that stores the variable or parameter once memory is assigned to it.

```

#include <stdio.h>
int a = 387;
int b;
void foo(int d) {
    int c;
    static int e;
    :
}

main() {
    static int f = 321;
    int g;
    :
}

```

- `a` - data segment
- `b` - BSS segment
- `c` - stack segment
- `d` - stack segment
- `e` - BSS segment
- `f` - data segment
- `g` - stack segment

Von Neumann machine

A variable in a program written in an imperative language is used to model what hardware component of a Von Neumann Architecture?

Memory cells.

What statement is used in an imperative language to model the piping operation of a Von Neumann machine?

Assignment statements.

Uncategorised

What are drawbacks of an ambiguous grammar?

Syntactic ambiguity of language structures is a problem because compilers often base the semantics of those structures on their syntactic form.

In particular, the compiler decides what code to generate for a statement by examining its parse tree. If a language structure has more than one parse tree, then the meaning of the structure cannot be determined uniquely.

(i) For an unambiguous grammar, how many parse trees do a derivation have? (ii) For an unambiguous grammar, could two different derivations be represented by the same parse tree?

- (i). Every derivation with an unambiguous grammar has a unique parse tree.
 - (ii). Yes, a parse tree can be represented by different derivations.
-

Other

Leftmost derivation

Parse tree

Prove the following grammar is ambiguous:

BNF