Linguaggi di programmazione - Modulo Kuper

1 Pointers and arrays

In C pointers and arrays are interchangeable

Equivalence and compatibility

given two tipes T and S are equivalent if every object of type T is also of type S.
T is compatible with S, if any object of type T can be used in a context where an object S is expected.

Structural equivalence

Compatibility

the definition depends on the language. S is compatible with T if:

- T and S are equivalent
- values of T are a subset of the values of S
- All operations on S can be performed on T
- there is a natural corrispondence
- values of T can be manipulated to correspond to some values of S

Type conversion

If T is compatable with S there is a type conversion mechanism:

- implicit conversion (or coercion): the abstract machine does the conversion
- explicit conversion (or cast): requires that the conversion is written in the program

Coercion

there are 3 types of coercion:

- same values and representation
- different values but common values have the same representation
- differet representation

Cast

the conversion is alowed only when the program knows how to do the conversion, and in certain casesmust be explicitly converted

Polymorphism

single value with multiple types:

- ad hoc Polymorphism (overloading)
- parametric Polymorphism

overloading

same simbols but with different meanings, resolved in compile time, after type inference

parametric Polymorphism

garbage collector

Dangling re