

CS 230
Programming Languages

10 / 17 / 2022

Instructor: Michael Eckmann

Today's Topics

- Questions? / Comments?
- Chapter 5
 - Names, Variables
 - Binding times
 - Code, static data area, stack and heap
 - Different kinds of variables in terms of when/where they are allocated
 - Scope
 - Lifetime

Names

- What's a name?
- A name (aka identifier) is associated with variables, labels, subroutine names, parameters, etc...
- What possible design criteria are there for names in a language? Any ideas?

Names

- Case sensitive? (Java, C++)
- Does it allow spaces or other whitespace? If so, are they ignored?
- What's the allowable length?
- What characters does it allow?
- What can it start with?
- Are words that are part of the language allowed as names?
 - those allowed as names are called **keywords**
 - those not allowed as names are called **reserved words**

Names

- some allow any length but only the first x characters are significant.
 - any problems or advantages to this?
- What about readability, writability, reliability, cost?
- Which of these ideas about names enhance or are a detriment to these language evaluation criteria?

Variables

- What's a variable?
- Is it the name of a memory location? Yes. But it is more.
- what else?

Variables

- What's a variable?
- Is it the name of a memory location? Yes. But it is more.
- Name, address, value, type, size
- And two more things:
- lifetime (how long the name is attached to the memory location),
- scope (where in the code can it be referenced)

Variables

- Memory address
 - may be possible for same name to be associated with different addresses
 - at different times during execution
 - at different places in the code
 - can anyone give an example of these?
 - the address of a variable is sometimes called its l-value (l for left side of an assignment statement)
- Aliases
 - anyone want to define alias in terms of names and addresses?

Variables

- Aliases
 - more than one name associated to the same memory address.
 - example: 2 variables x and y both refer to same address, so if I change the value that x refers to, the value that y refers to changes too because it contains the same address
 - effects on readability or writability?
 - An example of aliases in Java are that they are created when passing certain kinds of parameters (e.g. a reference to a class, or an array reference) Can aliases occur anywhere else in Java?
- Type
 - What does a type do?
- Size
- Value
 - sometimes referred to as a variable's r-value. Why do you think?

Variables

- Type
 - determines the kind and range of possible values and the operations that can be performed on variables of the type
- Size
 - how much memory --- determined by the type
- Value
 - contents of the variable at any given time during run-time
 - sometimes referred to as a variable's r-value. Why do you think?

Binding

- Binding is an association between an attribute and an entity
 - e.g. between a variable and its type
 - a variable and its value
 - an operation and a symbol
- Times when binding takes place
 - Language-design (aka compiler-design) time
 - Language implementation (aka compiler implementation) time
 - Compile time
 - Link time (when your compiled code is linked to other libraries)
 - Load time (when the program is executed it first gets loaded into memory)
 - Run time (when the program starts executing after being loaded into memory)

Binding

- Let me read a few examples from the text about when things are bound.

Binding

- Example from the text illustrates some of these ideas:

```
int count;
```

```
// ...
```

```
count = count + 5;
```

- The text discusses these assuming the code is in C++. Even if we don't know C++, we should be able narrow down when the bindings below occur.
- type of count is bound when?
- set of possible values of an int is bound when?
- meaning of + operator is bound when?
- internal representation of literal 5 is bound when?
- the value of count is bound when?

Binding

- The answers for the C language are:
- type of count is bound when? **compile time**
- set of possible values of int is bound when?
language-implementation time (when a compiler is created)
 - **what does that imply?**
- meaning of + operator is bound when? **compile-time**
 - **what does that imply?**
- internal representation of literal 5 is bound when
language-design time
- the value of count is bound when? **run-time**

Binding

- Examples of bindings that occur at
 - link time
 - a method/function call bound to the specific code in the library in which it is defined
 - load time
 - global variable may be bound to specific memory storage at load time
 - static variables in Java

Binding

- Binding attributes to variables
 - Static binding – first occurs *before* run time and stays same throughout program execution
 - note: load-time is considered before run-time
 - Dynamic binding
 - Either it first occurs *during* run time
 - OR it can change during execution