languages, whitespace, identifiers

lecture 3, mon jan 11 2010

borrowing from slides by kurt eiselt, wolfgang heidrich, alan hu

http://www.cs.ubc.ca/~tmm/courses/111-10
High-Level Language

- Must be translated into machine language so the computer can understand it.
- High-level instruction: \( A = B + C \) becomes at least four machine language instructions!

Interpreters and Compilers

- An interpreter translates the high-level language into machine language on-the-fly, executing the instructions as it goes.
- A compiler translates the high-level language program all at once in advance.
- Both compilers and interpreters are themselves computer programs.
- Which is better?
  - Remember George and Stephen in France?

Sample Java Application Program

```
/*   Oreo.java        Author:  Kurt Eiselt   Demonstrating simple Java programming ... void main (String[] args)  {    System.out.println ("Feed me more Oreos!");  }}

Sample Java Application Program

Comments ignored by Java compiler

```

Sample Java Application Program

```
Comments could also look like this

```

Sample Java Application Program

```

Sample Java Application Program

```

Sample Java Application Program

```

Sample Java Application Program

```

Reserved Words

- Get familiar with these
  - But you don’t need to memorize all 52 for exam

```

Identifiers

- Identifier must
  - Start with a letter and be followed by
  - Zero or more letters and/or digits
  - Digits are 0 through 9.
  - Letters are the 26 characters in English alphabet
    - both uppercase and lowercase
    - plus the $ and _
    - also alphabetic characters from other languages
  - Which of the following are not valid identifiers?

```java
public class Oreo {  
    public static void main(String[] args) {  
        System.out.println("Feed me more Oreos!");  
    }  
}
```

White Space

```java
//*******************************************************
// Oreo2.java       Author:  Kurt Eiselt
// Demonstrating case sensitivity of identifiers,
// which is important in a computer world.

public class Oreo2 {  
    public static void main(String[] args) {  
        System.out.println("Feed me more Oreos!");  
    }  
}
```

White Space

```java
//*******************************************************
// Oreo3.java        Author:  Kurt Eiselt
// Demonstrating totally bizarre uses of white space,
// since we should not be using them.

public class Oreo3 {  
    public static void main(String[] args) {  
        System.out.println("Feed me more Oreos!");  
    }  
}
```

Semantics

- What will happen when statement is executed
  - Programming languages have well-defined semantics, no ambiguity
  - Different than natural languages like English. Consider statement: Mary counted on her computer.

Syntax

- Rules to dictate how statements are constructed
  - Example: open bracket needs matching close bracket
  - If program is not syntactically correct, cannot be translated by compiler
  - Different than humans dealing with natural languages like English. Consider statement with incorrect syntax (grammar)
    - for weeks, rained in Vancouver it hasn’t
    - we still have pretty good shot at figuring out meaning

Identifiers

- Java is case sensitive
  - Oreo oreo OREO Oreo
  - are all different identifiers, so be careful
  - common source of errors in programming
  - are these all valid identifiers?
Errors

- Computers follow our instructions exactly.
- If program produces the wrong result it's the programmer's fault.
- You can't expect program to output correct results if you didn’t follow syntax rules that say how Java elements must be combined to form valid Java statements.
- Debugging: process of finding and correcting errors.
- Unfortunately can be very time consuming!

Memory and Identifiers

- Variable: name for location in memory where data is stored.
- Like variables in algebra class.
- Java convention, not compiler/syntax requirement.
- Variable names begin with lower case letters.
- More about that next time.

Programming With Variables

```java
//*****************************************
// Test.java       Author: Kurt
// Our first use of variables!
public class Test
{
    //variable declarations
    b = 3;
    c = 5;
    a = b + c;
    System.out.println ("The answer is " + a);
}
```

- Let's give it a try!
- b and c cannot be found!
- need to assign values

Memory Recap

- Memory: series of locations, each having a unique address.
- Locations in memory are not actually called things like a, b, and c.
- Instruction set tells computer to go to main memory and find value stored in location called A.
- Great! But... in reality, locations in memory are not actually called things like a, b, and c.
- Like variables in algebra class.
- High-level languages easy.
- Machine language uses addresses for memory locations.
- So what’s with the a, b, and c?
- Machine language uses addresses for memory locations.
- Variables begin with lower case letters.
- Memory: series of locations, each having a unique address, used to store programs and data.
- When data is stored in a memory location, previously stored data is over-written and destroyed.
- Each memory location stores one byte (8 bits) of data.
- Data values are stored in memory locations – more than one location may be used if the data is large.
- For total accuracy, these addresses should be binary numbers, but you get the idea, no?

Memory and Identifiers: Variables

- Variable: name for location in memory where data is stored.
- Like variables in algebra class.
- `pay_rate, hours_worked, a, b, and c` are all variables.
- Variable names begin with lower case letters.
- Java convention, not compiler/syntax requirement.
- Variable may be name of single byte in memory or may refer to a group of contiguous bytes.
- More about that next time.

Errors

- Error at compile time (during translation)
  - You did not follow syntax rules that say how Java elements must be combined to form valid Java statements.
  - A = B + C
  - Tells computer to go to main memory and find value stored in location called B
  - go to main memory and find value stored in location called C
  - add those two values together
  - store result in memory location called A
  - Great! But... in reality, locations in memory are not actually called things like a, b, and c.
  - Like variables in algebra class.
  - High-level languages easy.
  - Machine language uses addresses for memory locations.
  - So what's with the a, b, and c?
  - Machine language uses addresses for memory locations.
  - Variables begin with lower case letters.

Now What?

- Java doesn't know how to interpret the contents of the memory location.
- Are they integers? Characters from the keyboard? Shades of gray? or....

Errors

- Error at compile time (during translation)
  - You did not follow syntax rules that say how Java elements must be combined to form valid Java statements.
  - A = B + C
  - Tells computer to go to main memory and find value stored in location called B
  - go to main memory and find value stored in location called C
  - add those two values together
  - store result in memory location called A
  - Great! But... in reality, locations in memory are not actually called things like a, b, and c.
  - Like variables in algebra class.
  - High-level languages easy.
  - Machine language uses addresses for memory locations.
  - So what's with the a, b, and c?
  - Machine language uses addresses for memory locations.
  - Variables begin with lower case letters.

Data Types

- Java requires that we tell it what kind of data it is working with.
- For every variable, we have to declare a `data type`.
- Java language provides eight `primitive data types`.
- i.e. simple, fundamental.
- For more complicated things, can use data types.
  - created by others provided to us through the Java libraries.
  - that we invent.
  - more soon - for now, let's stay with the primitives.
  - We want a, b, and c to be integers.
  - Here's how we do it...

Programming With Variables: Take 2

```java
//*****************************************
// Test2.java       Author: Kurt
// Our second use of variables!
public class Test2
{
    //variable declarations
    b = 3;
    c = 5;
    a = b + c;
    System.out.println ("The answer is " + a);
}
```

- Source code compiles
- Object code runs
- But program may still produce incorrect results because logic of your program is incorrect.
- Typically hardest problems to find.

Errors

- Error at compile time (during translation)
  - You did not follow syntax rules that say how Java elements must be combined to form valid Java statements.
  - A = B + C
  - Tells computer to go to main memory and find value stored in location called B
  - go to main memory and find value stored in location called C
  - add those two values together
  - store result in memory location called A
  - Great! But... in reality, locations in memory are not actually called things like a, b, and c.
  - Like variables in algebra class.
  - High-level languages easy.
  - Machine language uses addresses for memory locations.
  - So what's with the a, b, and c?
  - Machine language uses addresses for memory locations.
  - Variables begin with lower case letters.

Programming With Variables: Take 3

```java
//*****************************************
// Test3.java       Author: Kurt
// Our third use of variables!
public class Test3
{
    //variable declarations
    int a; //these
    int b; //these
    int c; //variable declarations
    int d;
    a = b + c;
    System.out.println ("The answer is " + a);
}
```

- Logical error
- Source code compiles
- Object code runs
- But program may still produce incorrect results because logic of your program is incorrect.
- Typically hardest problems to find.

Errors

- Error at compile time (during translation)
  - You did not follow syntax rules that say how Java elements must be combined to form valid Java statements.
  - A = B + C
  - Tells computer to go to main memory and find value stored in location called B
  - go to main memory and find value stored in location called C
  - add those two values together
  - store result in memory location called A
  - Great! But... in reality, locations in memory are not actually called things like a, b, and c.
  - Like variables in algebra class.
  - High-level languages easy.
  - Machine language uses addresses for memory locations.
  - So what's with the a, b, and c?
  - Machine language uses addresses for memory locations.
  - Variables begin with lower case letters.
### Primitive Data Types: Numbers

#### Six primitives for numbers
- Integer vs. floating point
- Fixed size, so finite capacity

#### Types, Sizes, Min, Max

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>1 byte</td>
<td>-128</td>
<td>127</td>
</tr>
<tr>
<td>short</td>
<td>2 bytes</td>
<td>-32,768</td>
<td>32,767</td>
</tr>
<tr>
<td>int</td>
<td>4 bytes</td>
<td>-2,147,483,648</td>
<td>2,147,483,647</td>
</tr>
<tr>
<td>float</td>
<td>4 bytes</td>
<td>approx -1.7E308 (15 sig. digits)</td>
<td>approx 1.7E308 (15 sig. digits)</td>
</tr>
<tr>
<td>double</td>
<td>8 bytes</td>
<td>approx -1.7E308 (15 sig. digits)</td>
<td>approx 1.7E308 (15 sig. digits)</td>
</tr>
</tbody>
</table>

### Primitive Data Types: Non-numeric

#### Character Type
- Named char
- Java uses the Unicode character set so each char occupies 2 bytes of memory.

#### Boolean Type
- Named boolean
- Variables of type boolean have only two valid values: true and false
- Often represents whether particular condition is true
- More generally represents any data that has two states: yes/no, on/off

### Primitive Data Types: Numbers

- Primary primitives are `int` and `double`
- Just worry about those for now

### Questions?