important terms from today’s lecture in no particular order:

<table>
<thead>
<tr>
<th>term</th>
<th>term</th>
<th>term</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>identifier</td>
<td>constant/named</td>
</tr>
<tr>
<td>application</td>
<td>block</td>
<td>constant</td>
</tr>
<tr>
<td>comment</td>
<td>statement</td>
<td>strongly typed</td>
</tr>
<tr>
<td>whitespace</td>
<td>parameter</td>
<td>language</td>
</tr>
<tr>
<td>indenting</td>
<td>reserved word</td>
<td>data type</td>
</tr>
<tr>
<td>block vs inline</td>
<td>case sensitive</td>
<td>primitive data type</td>
</tr>
<tr>
<td>comment</td>
<td>declaration</td>
<td>literal</td>
</tr>
<tr>
<td>class</td>
<td>assignment</td>
<td>String literal</td>
</tr>
<tr>
<td>method</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>
• In the Java programming language:
  ▪ A program is made up of one or more classes
  ▪ A class contains one or more methods
  ▪ A method contains program statements

Java Program Structure

```java
// comments about the class
public class MyProgram {
    // class header

    class body

} // end class
```

Comments can be placed almost anywhere
Java Program Structure

// comments about the class
public class MyProgram
{

    // comments about the method
    public static void main (String[] args)
    {
        method body
    }
}

Basic program – heading – inline comments
Class – name begins with capital letter
Inline comments
Comment for overall program and the method
Every statement ends in a semi-colon.
Every statement in java has a particular syntax. See Appendix L for complete syntax.

public class Lincoln
{
    // Demonstrates the basic structure of a Java application.
    //****************************************************************
    // Lincoln.java       Author: Lewis/Loftus
    //****************************************************************
    public static void main (String[] args)
    {
        System.out.println ("A quote by Abraham Lincoln:");
        System.out.println ("Whatever you are, be a good one.");
    }
}
Second example is not formatted for readability

//****************************************************************************
//  Lincoln2.java       Author: Lewis/Loftus
//
//  Demonstrates a poorly formatted, though valid, program.
//****************************************************************************

public class Lincoln2{
  public static void main(String[] args){
    System.out.println("A quote by Abraham Lincoln:");
    System.out.println("Whatever you are, be a good one.");
  }
}
Third example incorporates our classroom style

//****************************************************************
//  Name: Lewis/Loftus
//  Date: 09/06/04
// Assignment: PianoKeys.java
//
//****************************************************************
//  PianoKeys class
//  Demonstrates the declaration, initialization, and use of an
//  integer variable.
//****************************************************************

public class PianoKeys
{
    /**------------------------------------------------------------
     //  Prints the number of keys on a piano.
     //  ---------------------------------------------------------*/
    public static void main (String[] args)
    {
        int keys;  // keys will hold the number of piano keys
        keys = 88;  // number of keys on a standard piano

        System.out.println ("A piano has " + keys + " keys.");
    }
}

Main method is described using javadoc format.
method name, main, is in lower case.
keys is a descriptive variable name in lower case.
Declaration associates a memory location with a name and a length (type).
Declaration and initialization are separated into two statements.
Everything within the class block is indented three spaces. Everything within the main block
is indented three additional spaces.
Final example shows more complex program using successive assignment.

```java
public class Geometry {
    /**
     * Demonstrates the use of an assignment statement to change the
     * value stored in a variable.
     */
    public static void main (String[] args) {
        int sides; // stores number of sides for the shape

        // Begin changing number of sides and printing shape.
        sides = 7;
        System.out.println ("A heptagon has " + sides + " sides.");

        sides = 10;
        System.out.println ("A decagon has " + sides + " sides.");

        sides = 12;
        System.out.println ("A dodecagon has " + sides + " sides.");
    }
}
```

**Declaration is separated from rest of the code.**  
**Each assignment print group is separated by whitespace.**  
**Program and main method are documented.**  
**Inline comments describe the variable, sides, and the actions to be carried out.**
Named constants

```java
public class PianoKeys4
{
    final int KEYS=88;  // since a constant does not change we declare and
    // initialize in same step

    System.out.println ("A piano has " + KEYS + " keys.");
}
```

Declare and initialize done as one step, only for CONSTANTS.
Note naming convention of constants. They may not appear to the left of an assignment after
first value assigned.
Keyword “final” designates KEYS as a constant.
Data and data types

**int** is a data type (can also be called an abstract data type). Used to hold integer values. It is a primitive type – it is built into the language.

Java has 6 different numeric primitive data types.

**int, byte, short, long, float, double**

See chart on page 74

Numeric representation of numbers means that the float numbers are approximation. Double has more precision than float.

Java has 2 other primitive data types.

**char, boolean**

All variables must be declared before use.

Java is strongly typed. If you declare a variable of a particular type, you cannot use it to hold a value of a different type.