CS1007 lecture #8 notes

tue 2 oct 2001

http://www.cs.columbia.edu/~sklar/cs1007

today:

- news
- command line arguments, one more time
- the `if` branching statement
- the `switch` branching statement
- type conversion
- compound `if` statement
- compound `switch` statement
- increment and decrement operators
- loops
- reading: *ch 2.6 3.1-3.5*

news.

midterm next class

- one page of notes allowed
- information link on web page

recitation sign-up — see me after class.

command line arguments, again.

```
import java.lang.*;
public class ex1 {
  public static void main ( String[] args ) {
    Integer tmp;
    int     i;
    System.out.println( "number of args = " + args.length );
    System.out.println( "first arg = " + args[0] );
    System.out.println( "second arg = " + args[1] );
    tmp = Integer.valueOf( args[0] ); // String -> Integer
    i = tmp.intValue();               // Integer -> int
    // now we're ready to do some Math!
  } // end of main
} // end of class ex1
```


the `if` branching statement.

```
import java.lang.*;
public class ex2 {
  public static void main ( String[] args ) {
    String s;
    char   c;
    System.out.println( "number of args = " + args.length );
    System.out.println( "first arg = " + args[0] );
    s = new String( args[0] ); // constructor
    c = s.charAt( 0 );
    System.out.println( "c = "+c );
    if ( c == '0' ) {
      System.out.println( "c is a digit" );
    }
    else if ( c == '1' ) {
      System.out.println( "c is a digit" );
    }
    else if ( c == 'A' ) {
      System.out.println( "c is a letter" );
    }
    else {
      System.out.println( "c is undefined" );
    }
  } // end of main
} // end of class ex2
```


the `switch` branching statement .

```
import java.lang.*;
public class ex3 {
  public static void main ( String[] args ) {
    String s;
    char   c;
    System.out.println( "number of args = " + args.length );
    System.out.println( "first arg = " + args[0] );
    s = new String( args[0] ); // constructor
    c = s.charAt( 0 );
    System.out.println( "c = "+c );
    switch( c ) {
    case '0':
      System.out.println( "c is a digit" );
      break;
    case '1':
      System.out.println( "c is a digit" );
      break;
    case 'A':
      System.out.println( "c is a letter" );
      break;
    default:
       System.out.println( "c is undefined" );
    } // end of switch
  } // end of main
} end of class ex3
```


`if` with compound statements .

```
import java.lang.*;
public class ex4 {
  public static void main ( String[] args ) {
    String s;
    char   c;
    System.out.println( "number of args = " + args.length );
    System.out.println( "first arg = " + args[0] );
    s = new String( args[0] ); // constructor
    c = s.charAt( 0 );
    System.out.println( "c = "+c );
    if (( c == '0' ) || ( c == '1' )) {
      System.out.println( "c is a digit" );
    }
    else if (( c == 'A' ) || ( c == 'B' )) {
      System.out.println( "c is a letter" );
    }
    else {
      System.out.println( "c is undefined" );
    }
  } // end of main
} // end of class ex4
```

compound `switch`.

```
import java.lang.*;
public class ex5 {
  public static void main ( String[] args ) {
    String s;
    char   c;
    System.out.println( "number of args = " + args.length );
    System.out.println( "first arg = " + args[0] );
    s = new String( args[0] ); // constructor
    c = s.charAt( 0 );
    System.out.println( "c = "+c );
    switch( c ) {
    case '0':
    case '1':
      System.out.println( "c is a digit" );
      break;
    case 'A':
      System.out.println( "c is a letter" );
      break;
    default:
      System.out.println( "c is undefined" );
    } // end of switch
  } // end of main
} // end of class ex5
```


increment and decrement operators.

- increment: ++

```
i++;
```

is the same as:

```
i = i + 1;
```

- decrement: --

```
i--;
```

is the same as:

```
i = i - 1;
```


looping.

- if you want to do something many times

- two types of loops:
— counter controlled
— condition controlled *(next time)*

- three loop statements:
— `for`
— `while`
— do

counter-controlled `for` loop.

```
import java.lang.*;
public class ex6 {
  public static void main ( String[] args ) {
    Integer tmp;
    int     n, i;
    tmp = Integer.valueOf( args[0] ); // String -> Integer
    n = tmp.intValue();               // Integer -> int
    System.out.println( "counting up to " + n + "..." );
    for ( i=0; i<n; i++ ) {
      System.out.print( i+ "  " );
    } // end for
    System.out.println();
  } // end of main
} // end of class ex6
```


counter-controlled `while` loop.

```
import java.lang.*;
public class ex7 {
  public static void main ( String[] args ) {
    Integer tmp;
    int     n, i;
    tmp = Integer.valueOf( args[0] ); // String -> Integer
    n = tmp.intValue();               // Integer -> int
    System.out.println( "counting up to " + n + "..." );
    i = 0;
    while ( i < n ) {
      System.out.print( i+ "  " );
      i++;
    } // end for
    System.out.println();
  } // end of main
} // end of class ex7
```


counter-controlled `do` loop.

```
import java.lang.*;
public class ex8 {
  public static void main ( String[] args ) {
    Integer tmp;
    int     n, i;
    tmp = Integer.valueOf( args[0] ); // String -> Integer
    n = tmp.intValue();               // Integer -> int
    System.out.println( "counting up to " + n + "..." );
    i = 0;
    do {
      System.out.print( i+ "  " );
      i++;
    } while ( i < n );
    System.out.println();
  } // end of main
} // end of class ex8
```