

CS1007 lecture #9 notes

tue 19 feb 2002

- news
- `System.exit()`
- assignment operators
- switch statement
- looping
- reading: ch 3.3, 3.5-3.8

cs1007-spring2002-sklar-lect09

1

news.

- MIDTERM #1 NEXT CLASS (THU 21 FEB)
 - you may bring one 8.5"×11" page of notes, both sides of page, any font
 - you MUST bring a photo ID
 - no calculators, laptops, palmtops, mobile phones, etc.
 - material covered:
 - * ch 1.1-1.5
(introduction; hardware components; networks; programming; programming languages)
 - * ch 2.1-2.7
(introduction to objects; string literals; variables and assignment; primitive data types and expressions; creating objects; class libraries and packages; invoking class methods)
 - * ch 3.1-3.8
(control flow; the if statement; the switch statement; boolean expressions revisited; more operators; the while statement; the do statement; the for statement)
- * plus ALL lectures and homeworks

cs1007-spring2002-sklar-lect09

2

System.exit()

- in class `java.lang.System`
- `public static void exit(int status);`
- terminates the currently running Java Virtual Machine.
- the argument serves as a status code — by convention, a nonzero status code indicates abnormal termination
- use at the end of a program to exit cleanly or to terminate in the middle

```
import java.lang.*;
public class ex9_exit {
    public static void main ( String[] args ) {
        if ( args.length < 4 ) {
            System.out.println( "usage: java hw2 <xr> <yr> <xb> <yb>" );
            System.exit( 1 ); // abnormal termination
        }
        // ... rest of program goes here ...
        System.exit( 0 ); // normal termination
    } // end of main()
} // end of class ex9_exit
```

cs1007-spring2002-sklar-lect09

3

increment and decrement operators.

- increment: `++`
`i++;`
is the same as:
`i = i + 1;`
- decrement: `--`
`i--;`
is the same as:
`i = i - 1;`

cs1007-spring2002-sklar-lect09

4

assignment operators.

```

+=
i += 3; is the same as: i = i + 3;

-=
i -= 3; is the same as: i = i - 3;

*=
i *= 3; is the same as: i = i * 3;

/=
i /= 3; is the same as: i = i / 3;

%=
i %= 3; is the same as: i = i % 3;

```

cs1007-spring2002-sklar-lect09

5

the if branching statement.

```

public class ex9_if1 {
    public static void main ( String[] args ) {
        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 ex9_if1

```

cs1007-spring2002-sklar-lect09

6

the switch branching statement .

```

public class ex9_switch1 {
    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 ex9_switch1

```

cs1007-spring2002-sklar-lect09

7

if with compound statements .

```

public class ex9_if2 {
    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' ) {
            System.out.println( "c is a letter" );
        }
        else {
            System.out.println( "c is undefined" );
        }
    } // end of main()
} // end of class ex9_if2

```

cs1007-spring2002-sklar-lect09

8

compound switch.

```
public class ex9_switch2 {
    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 ex9_switch2
```

cs1007-spring2002-sklar-lect09

9

looping.

- if you want to do something many times
- two types of loops:
 - counter controlled
 - condition controlled
- three loop statements:
 - for
 - while
 - do

cs1007-spring2002-sklar-lect09

10

counter-controlled for loop.

**** Note, as discussed in the afternoon class: `Math.random()` returns a value between 0.0 inclusive and 1.0 exclusive, so in order to get an integer between 0 and 51, we multiply the output from `Math.random()` by 52 and truncate it.**

```
public class ex9_for1 {
    public static void main ( String[] args ) {
        int n = 10, count;
        int card=(int)(Math.random()*52);
        for ( count=1; count<=n; count++ ) {
            card=(int)(Math.random()*52);
            System.out.println( "count="+count+" card="+card );
        } // end for
        System.out.println( "DONE!! count="+count+" card="+card );
        System.exit( 0 );
    } // end of main
} // end of class ex9_for1
```

cs1007-spring2002-sklar-lect09

11

counter-controlled while loop.

```
public class ex9_while1 {
    public static void main ( String[] args ) {
        int n = 10, count = 1;
        int card=(int)(Math.random()*52);
        while ( count<=n ) {
            System.out.println( "count="+count+" card="+card );
            card=(int)(Math.random()*52);
            count++;
        } // end while
        System.out.println( "DONE!! count="+count+" card="+card );
        System.exit( 0 );
    } // end of main
} // end of class ex9_while1
```

cs1007-spring2002-sklar-lect09

12

counter-controlled do loop.

```
public class ex9_d01 {
    public static void main ( String[] args ) {
        int n = 10, count = 1;
        int card=(int)(Math.random()*52);
        do {
            System.out.println( "count="+count+" card="+card );
            card=(int)(Math.random()*52);
            count++;
        } while ( count<n );
        System.out.println( "DONE!! count="+count+" card="+card );
        System.exit( 0 );
    } // end of main
} // end of class ex9_d01
```

cs1007-spring2002-sklar-lect09

13

condition-controlled for loop.

```
public class ex9_for2 {
    public static void main ( String[] args ) {
        int card1=(int)(Math.random()*52);
        int card2=(int)(Math.random()*52);
        int count=1;
        for ( ; card1 != card2; ) {
            System.out.println( "count="+count+" card1="+card1+" card2="+card2 );
            card1=(int)(Math.random()*52);
            card2=(int)(Math.random()*52);
            count++;
        } // end for
        System.out.println( "MATCH! count="+count+" card1="+card1+" card2="+card2 );
        System.exit( 0 );
    } // end of main
} // end of class ex9_for2

OR

.
.
.
for ( ; card1 != card2; card1=(int)(Math.random()*52),card2=(int)(Math.random()*52), count++ ) {
    System.out.println( "count="+count+" card1="+card1+" card2="+card2 );
} // end for
.
.
.
```

cs1007-spring2002-sklar-lect09

14

condition-controlled while loop.

```
public class ex9_while2 {
    public static void main ( String[] args ) {
        int card1=(int)(Math.random()*52);
        int card2=(int)(Math.random()*52);
        int count=1;
        while ( card1 != card2 ) {
            System.out.println( "count="+count+" card1="+card1+" card2="+card2 );
            card1=(int)(Math.random()*52);
            card2=(int)(Math.random()*52);
            count++;
        } // end while
        System.out.println( "MATCH! count="+count+" card1="+card1+" card2="+card2 );
        System.exit( 0 );
    } // end of main
} // end of class ex9_while2
```

cs1007-spring2002-sklar-lect09

15

condition-controlled do loop.

```
public class ex9_d02 {
    public static void main ( String[] args ) {
        int card1=(int)(Math.random()*52);
        int card2=(int)(Math.random()*52);
        int count=1;
        do {
            System.out.println( "count="+count+" card1="+card1+" card2="+card2 );
            card1=(int)(Math.random()*52);
            card2=(int)(Math.random()*52);
            count++;
        } while ( card1 != card2 );
        System.out.println( "MATCH! count="+count+" card1="+card1+" card2="+card2 );
        System.exit( 0 );
    } // end of main
} // end of class ex9_d02
```

cs1007-spring2002-sklar-lect09

16