



OBJECT ORIENTED PROGRAMMING

Abdullah Bin Kasem Bhuiyan
Topic 4: Lab Session

Listing: A program allocating, initializing, and printing `boolean` variables.

```
/** Examples for boolean variables */
public class BooleanVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        boolean var; // allocate boolean variable "var"

        var = true; // set variable var to true
        System.out.println(var); // prints true

        var = false; // set variable var to false
        System.out.println(var); // prints false

        var = false; // set variable var to false
        var = true; // set variable var to true
        System.out.println(var); // prints true

        boolean a = false, b = true, c = false; // allocate and initialize three variables
        System.out.println(a); // print the value of a, which is false
        System.out.println(b); // print the value of b, which is true
        System.out.println(c); // print the value of c, which is false
    }
}
```

Listing: A program allocating, initializing, and printing **byte** variables.

```
/** Examples for byte variables */
public class ByteVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        byte var; // allocate byte variable "var"

        var = -1; // set variable var to -1
        System.out.println(var); // prints -1

        var = -128; // set variable var to -128
        System.out.println(var); // prints -128

        var = 1_2_7; // set variable var to 127
        System.out.println(var); // prints 127

        byte hex = 0x10; // set hex to hexadecimal 10, which is 1*16+0 = 16: starts with "0x"
        System.out.println(hex); // prints 16

        byte bin = 0b0110_1111; // set bin to binary 01101111, which is 1+2+4+8+32+64=111: starts with
                               "0b"
        System.out.println(bin); // prints 111
    }
}
```

Listing: A program allocating, initializing, and printing `short` variables.

```
/** Examples for short variables */
public class ShortVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        short var; // allocate short variable "var"

        var = -1; // set variable var to -1
        System.out.println(var); // prints -1

        var = -32768; // set variable var to -32768
        System.out.println(var); // prints -32768

        var = 3_2767; // set variable var to 32767
        System.out.println(var); // prints 32767

        short hex = 0x10; // set hex to hexadecimal 10, which is 1*16+0 = 16: starts with "0x"
        System.out.println(hex); // prints 16

        short bin = 0b0110_1111; // set bin to binary 01101111, which is 1+2+4+8+32+64=111: starts with
            "0b"
        System.out.println(bin); // prints 111
    }
}
```

Listing: A program allocating, initializing, and printing `long` variables.

```
/** Examples for long variables */
public class LongVariables {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        long var; // allocate long variable "var"

        var = -1L; // set variable var to -1; notice the "L" which marks "long" literals for the compiler
        System.out.println(var); // prints -1

        var = -9223372036854775808L; // set variable var to -9223372036854775808
        System.out.println(var); // prints -9223372036854775808

        var = 9_223_372_036_854_775_807L; // set variable var to 9223372036854775807
        System.out.println(var); // prints 9223372036854775807

        long hex = 0x10L; // set hex to hexadecimal 10, which is 1*16+0 = 16: starts with "0x"
        System.out.println(hex); // prints 16

        long bin = 0b0110_1111L; // set bin to binary 01101111, which is 1+2+4+8+32+64=111: starts with "0b"
        System.out.println(bin); // prints 111
    }
}
```

Listing: A program allocating, initializing, and printing float variables.

```
/** Examples for float variables */
public class FloatVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        float var; // allocate float variable "var"

        var = -1f; // set variable var to -1; notice the "f" marking the "float" literal for the compiler
        System.out.println(var); // prints -1.0

        float fraction = 0.8f; // allocate and set variable fraction to 0.8
        System.out.println(fraction); // prints 0.8

        var = 1.4e-45f; // set variable var to  $1.4 \times 10^{-45}$ , the "aaaExxx" means "aaa*10xxx"
        System.out.println(var); // prints 1.4E-45

        var = 3.4028235e38f; // set variable var to  $3.4028235 \times 10^{38}$ 
        System.out.println(var); // prints 4028235E38

        float pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286f; // set
            pi to, well,  $\pi$ 
        System.out.println(pi); // prints 3.1415927 <- precision of float is about 7 decimals
    }
}
```

Listing: A program allocating, initializing, and printing double variables.

```
/** Examples for double variables */
public class DoubleVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        double var; // allocate double variable "var"

        var = -1d; // set variable var to -1; notice the "d" marking the "double" literal for the compiler
        System.out.println(var); // prints -1.0

        double fraction = 0.8d; // allocate and set variable fraction to 0.8
        System.out.println(fraction); // prints 0.8

        var = 4.9e-324d; // set variable var to  $4.9 \times 10^{-324}$ , the "aaaEzzz" means " $aaa \times 10^{zzz}$ "
        System.out.println(var); // prints 4.9E-324

        var = 1.7976931348623157e308d; // set variable var to  $1.7976931348623157 \times 10^{308}$ 
        System.out.println(var); // prints 1.7976931348623157E308d

        double pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286d; // set
            pi to, well,  $\pi$ 
        System.out.println(pi); // prints 3.141592653589793 <- precision of double is about 15 decimals
    }
}
```

Listing: A program allocating, initializing, and printing `char` variables.

```
/** Examples for character variables */
public class CharVariables {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        char var; // allocate character variable "var"

        var = 'T'; // set variable var to 'T'
        System.out.println(var); // prints 'T' (without the primes "'")

        var = '\u597d'; // set variable var to unicode char 0x597d
        System.out.println(var); // prints the Chinese character for "good"

        var = '\n'; // set variable var to literal \n, which stands for newline
        System.out.println(var); // prints a newline, i.e., an empty line

        char space = ' '; // set space to a space character
        System.out.println(space); // prints ' ' (without the primes "'")

        var = '\''; // setting var to ', using escaped single quote
        System.out.println(var); // prints '
    }
}
```

Listing: A program allocating, initializing, and printing String variables.

```
/** Examples for String variables */
public class StringVariables {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        String var; // allocate String variable "var"

        var = "Hello World!"; // set variable var to "Hello World!" //$NON-NLS-1$
        System.out.println(var); // prints "Hello World!" (without the quotation marks)

        var = "Hello\nWorld!"; // set variable var to "Hello\nWorld!" //$NON-NLS-1$
        System.out.println(var); // prints Hello, newline, World!

        String niHao = "\u4f60\u597d"; // set variable var to Hello in Chinese //$NON-NLS-1$
        System.out.println(niHao); // prints a "Ni Hao" in Chinese for Hello

        var = "\"Hello\""; // set variable var to "Hello" (using escaped double quotes)
        // $NON-NLS-1$
        System.out.println(var); // prints a "Hello" (without the quotation marks)
    }
}
```

Listing: The limits of the type `byte`.

```
/** Examples for the limits of byte variables */
public class ByteLimits {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        byte var; // allocate byte variable "var"

        var = Byte.MIN_VALUE; // set variable var to the minimum byte value
        System.out.println(var); // prints -128

        var = Byte.MAX_VALUE; // set variable var to the maximum byte value
        System.out.println(var); // prints 127

        var = Byte.SIZE; // set variable var to the size in bits of 1 byte
        System.out.println(var); // prints 8
    }
}
```

Listing: The limits of the type `short`.

```
/** Examples for the limits of short variables */
public class ShortLimits {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        short var; // allocate short variable "var"

        var = Short.MIN_VALUE; // set variable var to the minimum short value
        System.out.println(var); // prints -32768

        var = Short.MAX_VALUE; // set variable var to the maximum byte value
        System.out.println(var); // prints 32767

        var = Short.SIZE; // set variable var to the size in bits of 1 short
        System.out.println(var); // prints 16
    }
}
```

Listing: The limits of the type `int`.

```
/** Examples for the limits of integer variables */
public class IntLimits {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        int var; // allocate int variable "var"

        var = Integer.MIN_VALUE; // set variable var to the minimum int value
        System.out.println(var); // prints -2147483648

        var = Integer.MAX_VALUE; // set variable var to the maximum int value
        System.out.println(var); // prints 2147483647

        var = Integer.SIZE; // set variable var to the size in bits of 1 int
        System.out.println(var); // prints 32
    }
}
```

Listing: The limits of the type `long`.

```
/** Examples for the limits of long variables */
public class LongLimits {

    /** The main routine
     * @param args
     *      we ignore this parameter for now */
    public static final void main(final String[] args) {
        long var; // allocate long variable "var"

        var = Long.MIN_VALUE; // set variable var to the minimum long value
        System.out.println(var); // prints -9223372036854775808

        var = Long.MAX_VALUE; // set variable var to the maximum long value
        System.out.println(var); // prints 9223372036854775807

        var = Long.SIZE; // set variable var to the size in bits of 1 long
        System.out.println(var); // prints 64
    }
}
```

Listing: The limits of the type `float`.

```
/** Examples for the limits of float variables */
public class FloatLimits {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        float var; // allocate long variable "var"

        var = Float.MIN_VALUE; // set variable var to the minimum positive! float value
        System.out.println(var); // prints 1.4E-45

        var = Float.MIN_NORMAL; // set variable var to the minimum normal float value
        System.out.println(var); // prints 1.17549435E-38

        var = Float.MAX_VALUE; // set variable var to the maximum float value
        System.out.println(var); // prints 3.4028235E38

        int size = Float.SIZE; // set variable size to the size in bits of 1 float
        System.out.println(size); // prints 32

        var = Float.NEGATIVE_INFINITY; // set variable var to negative infinity
        System.out.println(var); // prints -Infinity

        var = Float.POSITIVE_INFINITY; // set variable var to positive infinity
        System.out.println(var); // prints Infinity

        var = Float.NaN; // set variable var to "not a number"
        System.out.println(var); // prints NaN
    }
}
```

Listing: The limits of the type `double`.

```
/** Examples for the limits of double variables */
public class DoubleLimits {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        double var; // allocate long variable "var"

        var = Double.MIN_VALUE; // set variable var to the minimum positive! double value
        System.out.println(var); // prints 4.9E-324

        var = Double.MIN_NORMAL; // set variable var to the minimum normal double value
        System.out.println(var); // prints 2.2250738585072014E-308

        var = Double.MAX_VALUE; // set variable var to the maximum double value
        System.out.println(var); // prints 1.7976931348623157E308

        int size = Double.SIZE; // set variable size to the size in bits of 1 double
        System.out.println(size); // prints 64

        var = Double.NEGATIVE_INFINITY; // set variable var to negative infinity
        System.out.println(var); // prints -Infinity

        var = Double.POSITIVE_INFINITY; // set variable var to positive infinity
        System.out.println(var); // prints Infinity

        var = Double.NaN; // set variable var to "not a number"
        System.out.println(var); // prints NaN
    }
}
```

Listing: Example for lossless compatible types.


```
/** Examples for lesslessly compatible variables */
public class CompatibleTypes1 {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        byte myByte = -128; // allocate byte variable "myByte"
        short myShort = myByte; // store value of myByte into variable myShort
        System.out.println(myShort); // prints -128

        myShort = 32767; // now store the mazimum short value in the variable
        int myInt = myShort; // and copy the value over to the new myInt variable
        System.out.println(myInt); // prints 32767

        myInt = 2147483647; // set myInt to the mazimum integer value
        long myLong = myInt; // and copy it to the new long variable myLong
        System.out.println(myLong); // prints 2147483647

        float myFloat = 1.4E-45f; // new float variable with the smallest positive float value
        double myDouble = myFloat; // copy its value into a double variable
        System.out.println(myDouble); // prints 1.401298464324817E-45, which is equivalent to 1.4E-45f
    }
}
```



It is possible to store values of an integer type into a floating point variable, but maybe with loss of information, i.e.

1 a byte value in a float variable (no loss of infos)

2 a short value in an float variable (no loss of infos)

3 an int value in a float variable (possible loss of infos)

4 a long value in a float variable (possible loss of infos)

5 a byte value in a double variable (no loss of infos)

6 a short value in an double variable (no loss of infos)

7 an int value in a double variable (no loss of infos)

8 a long value in a double variable (possible loss of infos)

Listing: Examples for the conversation of Integer Types to float.

```
/** Examples for int-to-float compatibility */
public class CompatibleTypes2Float {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        byte myByte = -128; // allocate byte variable "myByte"
        float myFloat = myByte; // store value of myByte into variable myFloat
        System.out.println(myFloat); // prints -128.0: no loss of infos, 8 bit fit into float mantissa

        short myShort = 32767; // now store the maximum short value in the variable
        myFloat = myShort; // and copy the value over to the new myFloat variable
        System.out.println(myFloat); // prints 32767.0: no loss of infos, 16 bit fit into float mantissa

        int myInt = 2147483646; // set myInt to the maximum integer value -1
        myFloat = myInt; // and copy it to the new long variable myLong
        System.out.println(myFloat); // prints 2.14748365E9, .e., rounding up to 2147483650

        long myLong = 9223372036854775806L; // set myLong to the maximum long value - 1
        myFloat = myLong; // and copy it to the new long variable myLong
        System.out.println(myFloat); // prints 9.223372E18, i.e., rounding down to 9223372000000000000L
    }
}
```

Listing: Examples for the conversation of Integer Types to `double`.

```
/** Examples for int-to-double compatibility */
public class CompatibleTypes2Double {

    /** The main routine
     * @param args
     *     we ignore this parameter for now */
    public static final void main(final String[] args) {
        byte myByte = -128; // allocate byte variable "myByte"
        double myDouble = myByte; // store value of myByte into variable myDouble
        System.out.println(myDouble); // prints -128.0: no loss of infos, 8 bit fit into double mantissa

        short myShort = 32767; // now store the maximum short value in the variable
        myDouble = myShort; // and copy the value over to the new myDouble variable
        System.out.println(myDouble); // prints 32767.0: no loss of infos, 16 bit fit into double mantissa

        int myInt = 2147483646; // set myInt to the maximum integer value -1
        myDouble = myInt; // and copy it to the new long variable myLong
        System.out.println(myDouble); // prints 2.147483646E9: no loss of infos, 32 bit fit into double mantissa

        long myLong = 9223372036854775806L; // set myLong to the maximum long value - 1
        myDouble = myLong; // and copy it to the new long variable myLong
        System.out.println(myDouble); // prints 9.223372036854776E18, i.e., rounding up to 9223372036854776000
    }
}
```