Dmitri Stanchevici
Unit 6
Module 1

Ex. 1.3

The compiler errors are

```
Ex_1_3.java:5: error: incompatible types: possible lossy conversion from int to byte
        byte a = 130;
               ^
Ex_1_3.java:8: error: incompatible types: possible lossy conversion from int to byte
            sum = sum + b;
                  ^
2 errors
```

The range of possible values of the byte type is -128 to 127. 130, to which byte a is initialized falls outside this range. This is even more the case for the values that sum=sum+b tries to produce.

Ex. 1.4

1. This program will compile.
2. The program will NOT execute with **java TestFileName**.
3. The compilation outputs the file called **A.class**.
4. To call the main() method inside class A from the main() method in public class TestFileName, I have to declare an instance of type A inside the TestFileName main() with a **new** operator and access this instance's main() method through the **. [dot]** operator, not forgetting to supply some parameters for the main() method inside A:

```
A a = new A();
String[] args = {"some", "arguments"};
a.main(args);
```

Ex. 1.6

Printed is

```
-126
```

The range of values of the **byte type is -128 to 127.** When 130 is explicitly cast as a byte value, it cycles back to the beginning of its range, -128, and gets increased by the difference between the int value cast as byte and the greatest possible byte value:

128 becomes -128
129 becomes -127
130 becomes -126

Ex. 1.11

The string

? || //**

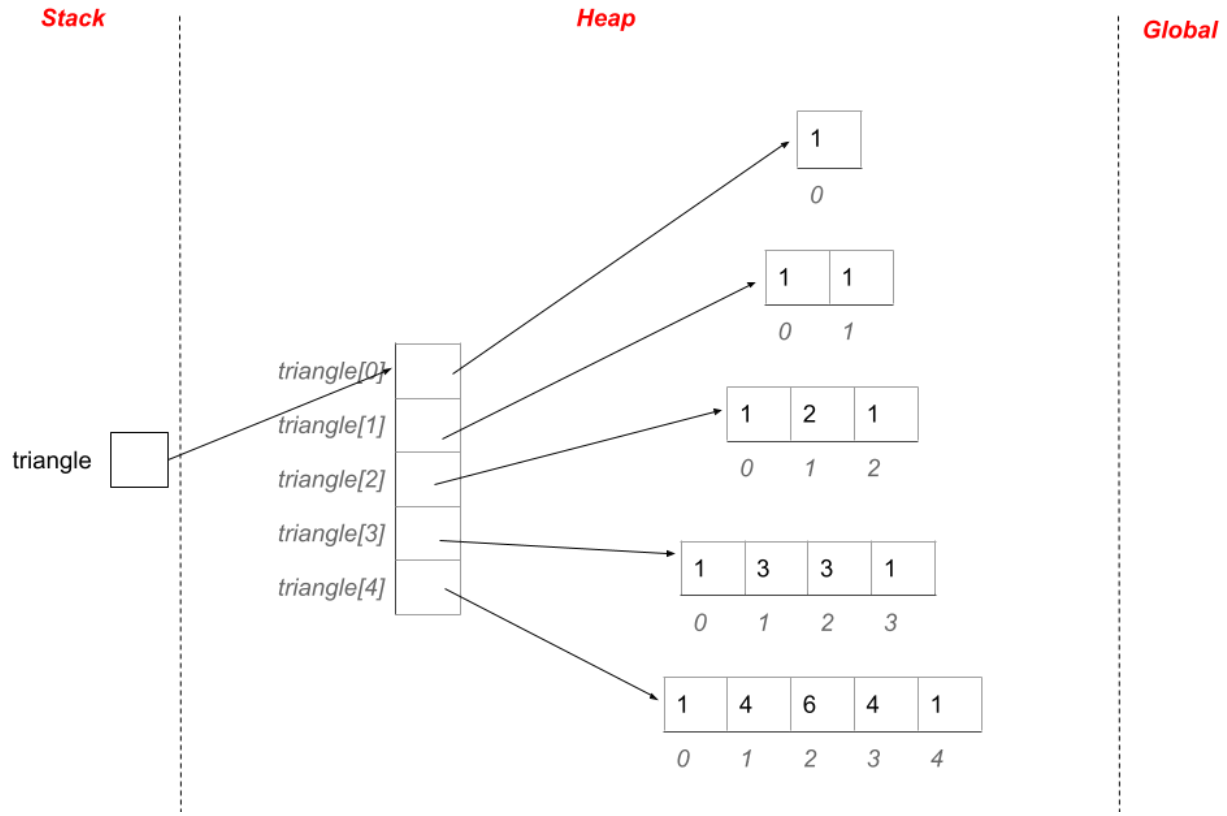
means “question or comment.”

Ex. 1.12

A compile error is printed:

```
Ex_1_12.java:6: error: incompatible types: int cannot be converted to boolean
        if (i = 4) {
            ^
1 error
```

Ex. 1.15



Ex. 1.19

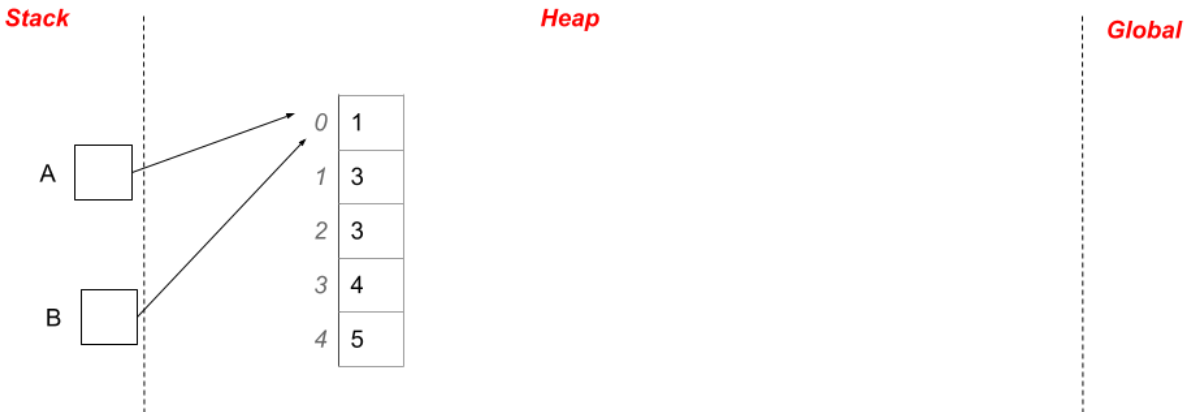
The output is

```
2
[1, 3, 3, 4, 5]
[1, 3, 3, 4, 5]
```

The call to **changeIt (B[0])** copied the value of **B[0]** into the parameter **int m** in the method's signature. It is this copy that has been changed and printed inside the method. The changes to this copy have no effect on the element of the array B[0].

The call to **changeItArray(B)** copied the address saved in the variable **B** into the parameter variable **int[] A** in the method's signature. Thus, these two variables started to point to the same array (to the same part of the heap memory containing the array). The elements of the array can now be accessed, modified, and retrieved via both **B** and **A**. Thus, even though **changeItArray**

(int[] A) is **void** (it does not return a value), any changes made to the array's values inside this method will be accessible through **B** in the **main** method.



Ex. 1.21

Call 1	Stack trace in add() return 3 Stack trace in main() int z = add (add(add(**1,2**), add(3,4)), 5)
Call 2	Stack trace in add() return 7 Stack trace in main() int z = add (add(3, add(**3,4**)), 5)
Call 3	Stack trace in add() return 10 Stack trace in main() int z = add (add(**3, 7**), 5)
Call 4	Stack trace in add() return 15 Stack trace in main() int z = add (**10, 5**)

Ex. 1.22

```
Ex_1_22.java:8: error: missing return statement
   }
   ^
1 error
```

Ex. 1.23

Yes, it is possible for a method to return a value that will not be used by the calling method.

Ex. 1.24

Two errors are thrown:

```
MethodExample3.java:18: error: method add(double,double) is already defined in class
MethodExample3
   static int add (double a, double b)
              ^
MethodExample3.java:20: error: incompatible types: possible lossy conversion from double to
int
          return (int) a+b;
                       ^
2 errors
```

The first error occurs because two methods have the same signature (the combination of name and the order and type(s) of the parameter(s): **add (double a, double b)**.

These methods have different return types and values, one **double** and the other **int**. However, the return type and value are not part of the method's signature; therefore, they cannot be used for overloading the method's name.

Ex. 1.25

Yes, the signatures of the given two methods are different because **the orders of the parameter types** are different in them.