Informatics I – EProg HS15

Exercise 2

1 Task: Primitive and Reference Types

1.1 Learning Objectives

1. Be able to explain the difference between primitive types and reference types.
2. Practice assignment operations with primitive types and reference types.

1.2 Assignment

a) Data Type Theory

Compare primitive types with reference types by completing table 1. You can use keywords.

	Primitive Type	Reference Type
Programming Language Integration		provided by the Java Standard Library (e.g.[1] String, Integer) or extensible with user-defined class definitions
Size in Memory		variable
Memory Layout (What is stored?)	the actual value	
Can have methods? (yes/no)		
Can store other data? (yes/no)		
Default Value	predefined but basically just zero (e.g. 0 for `int`, false for `boolean`)	

Table 1: Data Type Comparison

b) Functioning

What is the output of the following code snippets? Explain your answer!

```java
int alpha = 5;
int beta = 2;

alpha = beta;
alpha++;

System.out.println(alpha);
System.out.println(beta);
```

Listing 1: Snippet 1

Assume that the class `Number` can save an integer number via `setNumber(int newNumber)` and output its current value to the console via `printNumber()`. What is the output of the following code snippets? Explain your answer!

```java
Number x = new Number();
Number y = new Number();
x.setNumber(6);
y.setNumber(8);

x = y;

x.printNumber();
y.printNumber();

x.setNumber(10);
y.setNumber(2);

x.printNumber();
y.printNumber();
```

Listing 2: Snippet 2

2 Task: Statements

2.1 Learning Objectives

1. Know elementary kinds of statements.

2.2 Assignment

Provide two different examples for each kind of statement from below:

a) Declaration

b) Initialization

Reuse the variables declared in a).

c) Send a Message

d) Combined Declaration and Initialization

3 Task: Code Comprehension

3.1 Learning Objectives

1. Learn how to understand a class and its behavior by using the Java API with the example of the `String` class.
2. Practice reading and understanding code.

3.2 Assignment

Which of the following code snippets are syntactically correct and what is the output in that case? Explain your answer!

You can use:

- Walter Savitch, Java: An Introduction to Problem Solving and Programming
- Java API

```java
String s;
s = "Java is great!";
System.out.println(s);
```

Listing 3: Snippet 1

```java
String s;
"Java is great!" = s;
System.out.println(s);
```

Listing 4: Snippet 2

```java
String s = "Welcome";
s.toUpperCase();
System.out.println(s);
```

Listing 5: Snippet 3

```java
System.out.println("hELLo".toUpperCase());
```

Listing 6: Snippet 4

```java
String s = "Welcome";
PrintStream t = new PrintStream("at university");
t = s;
System.out.println(t);
```

Listing 7: Snippet 5

```java
String s = System.out.println("Welcome");
```

Listing 8: Snippet 6

```java
String s = new "Hello";
System.out.println(s);
```

Listing 9: Snippet 7

```java
String s = "h" + "ell".toUpperCase() + "o";
System.out.println(s);
```

Listing 10: Snippet 8

```java
String s = new String("hello".toUpperCase);
System.out.println(s);
```

Listing 11: Snippet 9

```java
String s;
String t = "Welcome";
s = "at university";
t = s;
s = t;
System.out.println(s);
System.out.println(t);
```

Listing 12: Snippet 10

```java
String s = new String("welcome").toUpperCase();
String t = "welcome".toUpperCase();
System.out.println(s);
System.out.println(t);
```

Listing 13: Snippet 11

4 Task: Cascading and Composition

4.1 Learning Objectives

1. Learn and practice cascading and composition of method calls with the example of String operations.

4.2 Assignment

Given the following declarations and initializations:

```
String s1 = "butterfly";
String s2 = "tiger";
```

a) Determine Result

What is the result of the following method calls?

1. `"gold".substring(1, 4).concat(s1.substring(6,7)).concat(s2.substring(2));`

2. `s1.substring(2).concat(s1.substring(2,3)).concat(s1.substring(1,3));`

b) Create Call Chain

Create call chains like above that produce the results below. You must only use the variables `s1`, `s2` and methods `substring()`, `concat()`. Note that multiple solutions are possible.

1. flyger

2. buttiger

5 Task: Object-oriented Concepts

5.1 Learning Objectives

1. Know the difference between classes and objects.
2. Understand the common terms of object-oriented concepts.

5.2 Assignment

1. Decide for each term of Table 2 which OO-concept (object-oriented-concept) is most appropriate. Is it a class, an object, a method, or an attribute?

Term	Class	Object	Method	Attribute
cat				
the cat "Garfield"				
color of the cat fur				
best friend of the cat				
the car "Herbie"				
car				
car number				
accelerate				
Porsche car				

Table 2: OO-Terms

2. Explain the following terms in the context of object-oriented programming:
   (a) Class

   (b) Instance

   (c) Message

   (d) Reference

   (e) Overloading

6 Task: Classes

6.1 Learning Objectives

1. You can implement a class that encapsulates state via attributes and behavior via methods.
2. You can verify the behavior of the implemented class with a TestDriver.

6.2 Assignment

a) Multiplier

Write a class with the name `Multiplier` and the methods `reset()`, `multiply()`, and `result()`.

- The default value of the result is 1.
- The method `multiply()` takes a single parameter of type `long` and multiplies the parameter by the result stored within the object.
- `result()` prints the result to the screen.
- `reset()` sets back the object by resetting the result to 1.

```java
public class Multiplier {

   . . .

   public void multiply( . . . ) {
      . . .
   }

   public void reset() {
      . . .
   }

   public void result() {
      . . .
   }
}
```

b) MultiplierTestDriver

Write a class called `MultiplierTestDriver` with a `main()` method wherein you create at least 3 `Multiplier` class instances. Test the methods `reset()`, `multiply()`, and `result()`. Also test the assignment operator (=) and call methods afterward (e.g. something like `multiplier1 = multiplier2` followed by `adder1.result()`).