Exam
DIT948: Programming

Date:	2016-08-22
Time	8:30–12:30
Place	Lindholmen
Teacher	Musard Balliu
Examiner	Ivica Crnkovic
Questions	031 772 1098
	first visit around 9:30
Results	will be posted within 15 working days
Grades	Pass (G) 50p, Pass with Honors (VG) 80p
Allowed aids	Any book on Java programming is allowed
	Any English dictionary
	No electronic translators are allowed
Reviewing:	2016-08-31, Lindholmen
	(Note: subject to change, time and place will
	be posted on the course website).

Please observe carefully the following:

- Write in legible English (illegible translates to “no points”!).
- Motivate your answers, and clearly state any assumptions made.
- Code or text copied from the book will not be rewarded. Your own contribution is required.
- Start each task on a new sheet!
- Write on only one side of the paper!
- Before handing in your exam, number and sort the sheets in task order!
- Write your anonymous code and page number on every page!

Not following these instructions will result in the deduction of points!

1. [10pts] Consider the following Java program:

```
import static dit948.SimpleIO.*;

public class PrintMe {
  private static String username = "user";
  private int password = 1234;

  public PrintMe(){}

  public PrintMe(String username){
    PrintMe.username = username;
    this.password = 5678;
  }

  public void changePassword(int password){
    println("The old password of " + this.getUsername() +
            " was " + this.password);
    this.password = password;
  }

  public String getUsername(){
    return PrintMe.username;
  }

  public static void main(String[] args){
    PrintMe o1 = new PrintMe();
    PrintMe o2 = new PrintMe("superUser");
    println(o1.getUsername());
    println(o1.password);
    println(o2.getUsername());
    println(o2.password);
    o1.changePassword(9000);
    println(o1.getUsername());
    println(o1.password);
    println(o2.getUsername());
    println(o2.password);
    }
}
```

What will be printed to the console (standard output) when running the program PrintMe?

2. [10pts] Your boss wrote a program that was supposed to print to the console the average and the maximum of a non-empty array of integers. The average is the sum of the integers divided by the number of integers in the array, and the maximum is the largest integer in the array.

   Unfortunately, things didn't really work out as expected: in fact, the program doesn't even compile! Still, it's your boss, you can't just throw it away and write a new program. Make the changes necessary for the program to compile and print the correct result.

```
public static int averageAndMax(int[] arr){
    double avg = 0;
    int max = 0;
    double sum = 0;
    for (int i=0; i<=arr.length; i++){
      sum +=arr[i];
      if (max > arr[i]){
        max = arr[i];
      }
    }
    avg = sum/arr.length;
    println("The average is "+ average);
    println("The maximum is "+ max);
}
```

3. [10pts] Consider the following classes:

```
public class A {
  private String s = "Hej";

  public String toString(){
    return this.s + " from class A";
  }
}


public class B extends A {
  private String s1 = "Hello";
  public String toString(){
    return this.s1 + " from class B";
  }
}
```

Which of the lines 3–12 in class `Test`, if any, will cause a compilation error and why? Comment out such lines (if any) and write what will be printed out to the console when running the class `Test`.

```
0 import static dit948.SimpleIO.*;

1 public class Test {
2    public static void main(String[] args) {
3      A a1 = new A();
4      println(a1);
5      A a2 = new B();
6      println(a2);
7      a2 = a1;
8      println(a2);
9      a1 = a2;
10     println(a2);
11     B b =  new A();
12     println(b);
13    }
14 }
```

4. [30pts]

The 2016 Summer Olympic program features 28 sports and a total of 306 events. In this exercise, you will implement two Java classes to represent different characteristics of a sport and an event.

A `Sport` is specified by the name, the country and the number of athletes from that country competing in that sport. For instance, *Sweden* has *11* athletes competing in the sport of *Aquatics*.

An `Event` extends a `Sport` by (additionally) specifying the sport, the name of the event, the name of the athlete and their placement in that event. For instance, the Swedish athlete *Sarah Sjöström* was placed number *1* in the event *Women's 100m butterfly* in the sport *Aquatics*.

You will implement the class Sport and its subclass Event by “filling in” the missing data and method definitions. You don’t need to do any error checking or write any comments.

Please make sure you do not write “between the lines”! On your paper, clearly mark out which code is supposed to go where. For example, the code supposed to fill in the block A, the instance variable of the class Sport, should appear as

```
+-------------A-------------+
|  private String sportName;  |
+-----------------------------+
```

and not squeezed on the page with the exam subjects!

Remarks:

- Since block A has been filled in for you above, there are 12 blocks left for you to fill in for this exercise. Blocks B, D and F are worth 1 point each, blocks C and I are worth 2 points each, blocks E, H, J, K and L are worth 3 points each, and blocks G and M are worth 4 points each.

End of remarks.

```
package olympics;

// Class specifying a Sport

public class Sport {
  // Block A:
  // Declare a private instance variable,
  // sportName, a string specifying the sport name

  // Block B:
  // Declare a private instance variable,
  // country, an string specifying the country

  // Declare a private instance variable,
  // athletes, an integer specifying the number of athletes

  public Sport(){
    // Block C:
    // Default constuctor that initializes the sport name to Aquatics,
    // the country to Sweden and the athletes’ number to 11
  }

  public Sport(String sportName, String country, int athletes){
    // Block D:
    // Constructor that initializes the instance variables
    // with the given arguments
  }

  public Sport(Sport sport){
    // Block E:
    // Constructor that initializes the instance variables with
    // the given argument.
  }

  public String getSportName(){
    // Block F:
    // Return the value of instance variable sportName
  }

  public boolean sameSport(Sport s){
    // Block G:
    // Given the current Sport (this) and another Sport s,
    // return true if the sport name is the same and both
    // countries have at least 1 athlete competing in that sport
  }

  public String toString(){
    // Block H:
    // Return the string representation of a sport
    // For example, if sportName = Aquatics, country = Sweden,
    // and athletes = 11, it returns the string
    // "Sweden has 11 athletes in Aquatics"
  }
}
```

```
package olympics;
// Class specifying an olympic event

public class Event extends Sport {
  // Block I:
  // Declare private instance variables,
  // sport, a Sport specifying the sport object
  // eventName, a string specifying the event name
  // placement, an integer specifying the placement
  // athleteName, a string specifying the athlete name

  public Event(){
    // Block J:
    // Default constructor that initializes the instance variables with
    // values of the default constructor in class Sport, and placement = 1,
    // eventName = Women’s 100m butterfly, athleteName = Sarah Sjostrom
  }

  public Event(Sport sport, String eventName, int placement,
                                        String athleteName){
    // Block K:
    // Constructor that initializes the instance variables
    // with the given arguments
  }

  public String toString(){
    // Block L:
    // Return the string representation of an event. For example, if
    // sportName = Aquatics, placement = 1, athleteName = Sarah Sjostrom,
    // eventName = Women’s 100m butterfly, it returns the string
    //"Sarah Sjostrom was ranked 1 in Women’s 100m butterfly in Aquatics"
  }

  public String compareTwoAthletes(Event e){
    // Block M:
    // Given the current Event (this) and another Event e,
    // compare the placements of the corresponding athletes and
    // return a string specifying who performed better.
    // Examples: "A performed better than B", "A performed the same as B"
  }
}
```

5. [20pts] In this exercise, we refer to the classes `Sport` and `Event` from above, but do not require you to have implemented them (only that you understand how to create and use instances of them).

   (a) In class `Sport`, implement a static method with the signature

   ```
   public static String maxAthletesJudo(Sport[] s)
   ```

   that returns a string specifying the country with the largest number of athletes in the sport of Judo. You can assume that the array has at least one element. In case there are several countries with the same number of Judo athletes and this is the largest number, you should return the first country with such number.

   For example, Brazil and Japan have 14 Judo athletes each and any other country have less. Then, if Brazil appears at position 0 of array s and Japan at position 3, you should return the string "Brazil has 14 athletes in Judo". Otherwise, if the array s does not contain the sportName Judo at all, you should return the string "No Judo players in this olympics".

   (b) In class `Event`, implement a method with the signature

   ```
   public static void firstThree(Event[] e){
   ```

   that prints to the console the winners of the first three places (in order from placement 1 to 3) for eventName = Women's and sportName=Golf, from the array of events e. Example of output to the console:

   ```
   Pernilla Lindberg was ranked 1 in Women's in Golf
   Miriam Nagl was ranked 2 in Women's in Golf
   Nicole Broch Larsen was ranked 3 in Women's in Golf
   ```

   Each method is worth 10 points.

6. [20pts] (From *CodingBat*)

   Write a subroutine takes as argument `start` and `end` integers, and returns a new array containing the sequence of integers from `start` up to but not including `end`. For insstance, `start`=5 and `end`=10 yields the array {5, 6, 7, 8, 9}. You can assume the `end` number is greater or equal to the `start` number. Note that an array of length 0 is also a valid aray.

Examples:

```
fizzArray3(5, 10) -> {5, 6, 7, 8, 9}
fizzArray3(11, 18) -> {11, 12, 13, 14, 15, 16, 17}
fizzArray3(1, 3) -> {1, 2}
fizzArray3(1, 1) -> {}
```