Exam
DIT948: Programming

Date:	2016-01-05
Time	8:30–12:30
Place	Lindholmen
Teacher	Musard Balliu
Examiner	Ivica Crnkovic
Questions	0762416640 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:	2016-01-22, Lindholmen Fourth floor room in Jupiter (Note: subject to change, changes 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 Ex1 {

  private String[] names = {"John","Maria"};
  private  int secret = 11;
  static int num = 2;

  public static void changeSecret(int secret, int newSecret){
     secret = newSecret;
     println("The value of  secret in changeSecret is: "+ secret);
     println("The value of  num in changeSecret is: "+ num);
     num=secret;
  }

  public static void swapNames(String[] names){
     String tmp;
     tmp = names[0];
     names[0]=names[1];
     names[1]=tmp;
     println("The value of names[0] in swapNames is "+ names[0]);
     println("The value of  num in swapNames is: "+ num);
  }

  public static void main(String[] args) {
     String[] names = {"Rickard", "Jennifer"};
     int secret = 22;
     Ex1 obj = new Ex1();
     println("Initial value of secret is: "+ secret);
     changeSecret(secret, obj.secret);
     swapNames(names);
     println("The value of names[0]  is "+ names[0]);
     swapNames(obj.names);
     println("The value of obj.names[0]  is "+ obj.names[0]);
     println("The value of num  is: "+ num);
     }
}
```

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

the program Ex1?

2. [10pts] Your boss wrote a program that was supposed to compute whether or not a natural number n (greater than 1) is a *prime* number. Recall that prime number is a natural number n greater than 1 that has no positive divisors other than 1 and n itself. For instance, 3 is a prime number since it can only be divided by either 1 or 3, while 4 is not a prime number since it can be divided by 2. Also recall that $a\%b$ is the remainder of the division of a by b; for instance $5\%3 = 2$ and $20\%2 = 0$.

   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 void isPrime(natural n){
   for(int i=2; i<=n;i++){
       if(n%i == 1){
          return false;
       }
       return true;
   }
}
```

3. [10pts] Consider the class:

```
package party;

public class Party {

    private  String host;
    private  int numGuests;
    boolean  bringOwnDrink;
    public double location;
    protected String[] guestlist;

    protected boolean bringOwnDrink() { return this.bringOwnDrink; }


    private void updateNumGuests(int newGuests) {
```

```
        this.numGuests += newGuests;
    }

    public String toString() {
        String str ="";
        for (int i=0; i<this.guestlist.length; i++){
           str += guestlist[i]+"\n";
        }
        return str;
   }

   String[] getGuestlist() { return this.guestlist; }

   String getHost() { return this.host; }
}
```

Class `ThrowParty` is a subclass of `Party`:

```
 1 package party;
 2
 3 public class ThrowParty extends Party {
 4     public void throwParty() {
 5         System.out.println(host);
 6         System.out.println(numGuests);
 7         System.out.println(bringOwnDrink);
 8         System.out.println(bringOwnDrink());
 9         System.out.println(location);
10         System.out.println(toString());
11         updateNumGuests(10);
12         System.out.println(getGuestlist());
13         System.out.println(getHost());
14     }
15 }
```

Which of the lines 5–13 will cause a compilation error and why? How does the answer change if `ThrowParty` is in a different package from `party` (for example, in a package `otherParty` and it replaces the first line with `import party.*;`)?

4. [30pts]

   In this exercise, you will implement several classes to represent different characteristics of a city.

   You will do this by "filling in" the data and method definitions. You don't need to do any error checking or write comments.

   Please 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 member variable of the class `City`, should appear as

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

   and not squeezed on the page with the exam subjects!

   Remarks:

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

   End of remarks.

```
// Class representing a city
public class City {
  // Block A:
  // Declare a private instance variable,
  // name, a String representing the city name

  // Block B:
  // Declare a private instance variable,
  // population, an integer representing the city population

  public City(){
    // Block C:
    // Initialize the city name to Goteborg
```

```
  // and the population to 500000
}

public City(String name, int population){
  // Block D:
  // Initialize the instance variables with
  // the given arguments
}

public City(City c){
  // Block E:
  // Initialize the instance variables with
  // the given argument
}

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

public int getPopulation(){
  // Block G:
  // Return the value of instance variable
  // population
}

public boolean sameCity(City otherCity){
  // Block H:
  // Return true if the name of the current (this) city
  // is the same as the name  of otherCity
}

public int totalPopulation(City otherCity){
  // Block I:
  // Return the total population of the current (this) city
  // and  otherCity
}

public String toString(){
  // Block J:
```

```
    // Return the string representation of a city
    // If name= Goteborg and population = 500000, it returns
    // "Goteborg has  a population of 500000 inhabitants."
  }
}


// Class representing a city and a league of
// east coast cities in Sweden

public class EastCoastCity extends City {
  // Block K:
  // Declare two private instance variables,
  // city, a City representing the city object
  // eastLeague, an array of City representing the
  // east coast league

  public EastCoastCity(City city, City[] eastLeague, String name,
                                                    int population) {
    // Block L:
    // Initialize the instance variables with
    // the given arguments
  }

  public EastCoastCity(){
    // Block M:
    // Initialize the name with Stockholm,
    // the population with 1000000, the city with
    // the object corresponding to Stockholm and
    // the eastLeague with an array containing
    // the cities of Stockholm and Uppsala
  }

  public boolean isEastLeague(City c){
    // Block N:
    // Return true if a city c is part of the
    // east coast league, namely if a city with
    // the same name is part of the league
  }
```

```
public String toString(){
  // Block 0:
  // Return the string representation of the east coast league, i.e.
  // "The east league consists of the cities: (city names here)"
}
```

5. [20pts] In this exercise, we refer to the classes 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 City, implement a method with the signature

   `public boolean differAtLeastTwoLetters(City otherCity)`

   that returns true if the name of the current city (this city) differs in at least two characters from the name of the otherCity. You can assume that the name strings have the same length (number of characters).

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

   `public EastCoastCity addToLeague(EastCoastCity ec){`

   that returns a new EastCoastCity object containing the same City object, name and population field as in ec and the eastLeague array field extended with the City object from ec.

   Each method is worth 10 points.

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

   Write a subroutine takes as argument an array of integers and returns an array that is *left shifted* by one – so $\{6, 2, 5, 3\}$ returns $\{2, 5, 3, 6\}$. You may modify and return the given array, or return a new array. Remember to check the corner cases! The subroutine should always return a value and not crash.

   Examples:

```
shiftLeft({6, 2, 5, 3}) -> {2, 5, 3, 6}
shiftLeft({1, 2}) -> {2, 1}
shiftLeft({1}) -> {1}
```