Data Structures and Algorithms  
Week 1 problem sheet

## A. Algorithms and Data Structures

1. In your own words, complete the following sentences.
   1. “An algorithm is … .”
   2. “A data structure is …. .”

* **Answer:**
* You should answer question 1 using your own words, so no model solution is provided. However, you might wish to review the slides from the first lecture.

## B. Java basics

1. Consider the following Java code:

* class Person {  
   private String name;  
   private int age;  
   Person bestFriend;  
    
   public Person(String name, int age, Person bestFriend) {  
   this.name = name;  
   this.age = age;  
   this.bestFriend = bestFriend;  
   }  
    
   public int getAge() {  
   return age;  
   }  
    
   public void setAge(int age) {  
   this.age = age;  
   }  
    
  }
* Answer the following questions about the code.
  1. What is the name of the class being defined here?
  2. Where is the constructor for the class? What does it do?
  3. What are the *instance variables* of the class?
  4. What are the *methods* of the class?
  5. What are the *parameters* of the setAge method? What are their *types*?
  6. What is the type of the bestFriend instance variable? In what ways is it different from the age instance variable?
* **Answer:**
  1. The name of the class is “Person”.
  2. The constructor of the class is the portion starting with the line
  + public Person(String name, int age, Person bestFriend)
  + It ensures that all the instance variables of Person are initialized and that the new Person object is in a consistent state.
  1. The instance variables are name, age, and bestFriend.
  2. The methods are getAge and setAge
  3. The only parameter of the setAge method is age, which has type int.
  4. The bestFriend instance variable is of type Person. It is different from age in that Person is a *reference* type, whereas int is a *value* type.

1. What kind of “thing” is each part of the following Java statement? How does it work?

* System.out.println(myObject);
* **Answer:**
* System is a class (you can find the documentation for it on the Oracle website, at <https://docs.oracle.com/javase/7/docs/api/java/lang/System.html>). out is a static field of the System class, and println is a method of out. myObject is an argument being passed to the println method; we are not told its type.

1. What is meant by each part of the following Java code fragment:

* public static void main(String[] args)
* **Answer:**
* - The public modifier means that the main method can be accessed by code in any class, rather than being limited to code in methods of the class that main is contained in.
* - The static modifier means the method can be called even without an object to invoke it on.
* - The void return type means the method does not return a value.
* - The portion within parentheses (“String[] args”) means that this method takes one argument called args, which is an array of Strings.

1. What is meant by each part of the Java statement:

* import java.util.ArrayList;
* How does it differ from the following?
* import java.util.\*;
* **Answer:**
* The first import statement imports the java.util.ArrayList class, meaning it can be referred to without having to specify its pacage.
* The second one imports *every* class in the java.util package.

1. Suppose we execute the following two Java statements:

* int a = 32;  
   int b = 32;
* What would then be the value of the expression “a == b”?
* **Answer:**
* The value of “a == b” would be true.

1. Suppose we execute the following two Java statements:

* java.awt.Color red = new java.awt.Color(255,0,0);  
   java.awt.Color xxx = new java.awt.Color(255,0,0);
* What would then be the value of the expression “red == xxx”, and why?
* **Answer:**
* The value of “a == b” would be false. This is because java.awt.Color is a *reference* type, not a value type (which we can tell, because it is constructed using the “new” keyword).
* When applied to reference types, the “==” (equality) operator compares the location in memory of the objects being compared, which in this case will not be the same; hence, the result is “false”.

1. We have two monkeys, A and B, and the boolean variables aSmile and bSmile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Which of the following expressions evaluates to true if and only if we are in trouble?
   1. aSmile || bSmile
   2. aSmile && bSmile
   3. aSmile == bSmile

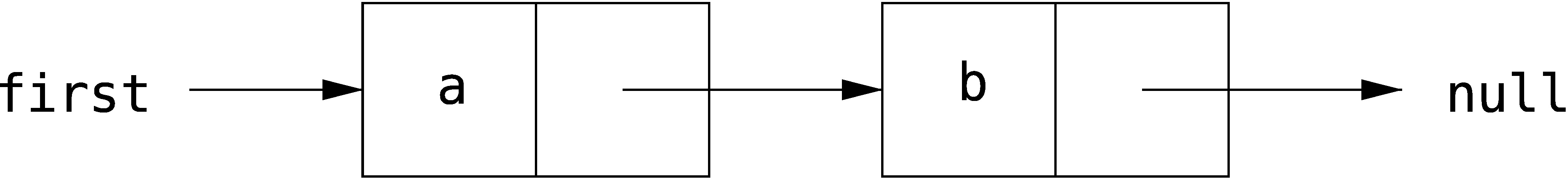
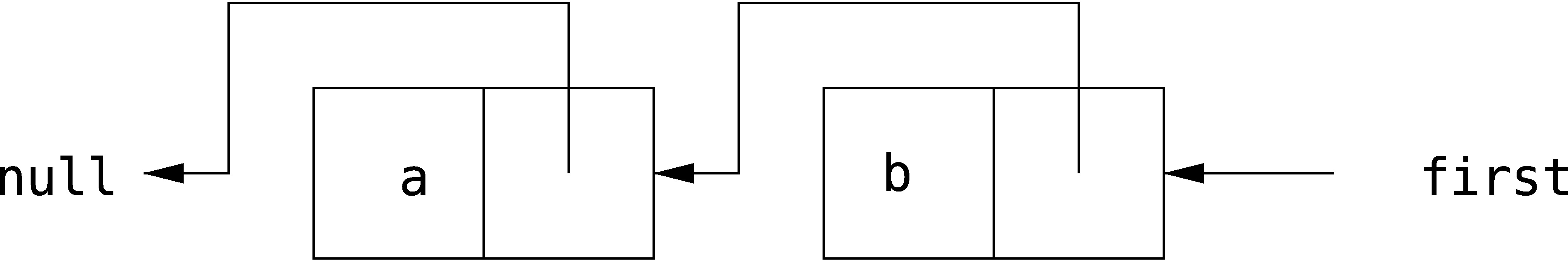
* **Answer:**
* The correct answer is **c**. This will evaluate to true if both aSmile and bSmile are false, and also if they are both true.

For more Java revision questions, see

[**http://teaching.csse.uwa.edu.au/units/CITS2200/Tutorials/tutorial02.html**](http://teaching.csse.uwa.edu.au/units/CITS2200/Tutorials/tutorial02.html)

## C. Linked Lists

1. Let Link be an object with two member variables:

* public class Link {  
   char item;  
   Link next;  
   }
* Assume that the variable first is a reference to a Link object containing 'a', whose next is a link object containing 'b', whose next is null.
* 
* Which of the following code snippets successfully reverses this structure, so as to give the following list?
* 
* Draw a picture of the result of each code snippet to explain your answer.
  1. first.next = null;  
     first.next.next = first;  
     first = first.next;
  2. first.next.next = first;  
     first = first.next;  
     first.next.next = null;
  3. Link temp = first;  
     first = temp.next;  
     first.next = temp;  
     temp = null;
  4. Link temp = first.next;  
     temp.next = first;  
     first.next = null;
* **Answer:**
* The correct answer is **b**.