

Object-Oriented Software Engineering (Re-exam for Object-Oriented Analysis, Design and Programming)

Medialogy, 4th Semester, Aalborg

Monday 11 June 2012, 09.00 – 12.00

Instructions

You have 3 hours to complete this examination.

Neither written material nor electronic equipment may be brought into the examination room.

There are 24 questions and each question is worth 5 marks. The maximum possible score is 120 marks. You must get at least 50 marks to pass.

The following table shows the grade that will be awarded for a given number of marks:

No. of Marks	Grade
0 – 24	-3
25 – 49	00
50 – 59	02
60 – 69	4
70 – 79	7
80 – 89	10
90 – 120	12

Note that a number at the beginning of any line in a program listing indicates the number of that line and is not part of the program itself.

DO NOT TURN OVER UNTIL INSTRUCTED TO DO SO!

Question 1

Which one of the following was the main cause of failure in the Ariane 5 project?

- A. Software inflexible and difficult to maintain.
- B. Software re-used without thoroughly testing its operation in the new context.
- C. Chose to build an expensive new system instead of modifying an existing one.
- D. Software interlocks did not prevent a dangerous configuration.

Question 2

Which one of the following was one of the main problems in the Taurus project?

- A. Software inflexible and difficult to maintain.
- B. Software re-used without thoroughly testing its operation in the new context.
- C. Chose to build an expensive new system instead of modifying an existing one.
- D. Software interlocks did not prevent a dangerous configuration.

Question 3

Write down the output of the following Java program.

```
public class WhileLoop {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i++;
        }
    }
}
```

Question 4

Write down the output of the following Java program.

```
public class CondPercent {
    public static void main(String[] args) {
        double x = -1.0;
        do {
            System.out.print(x % -1);
            if (x % 1 == 0)
                System.out.println(" int");
            else
                System.out.println(" float");
            x += .5;
        } while (x < 1);
    }
}
```

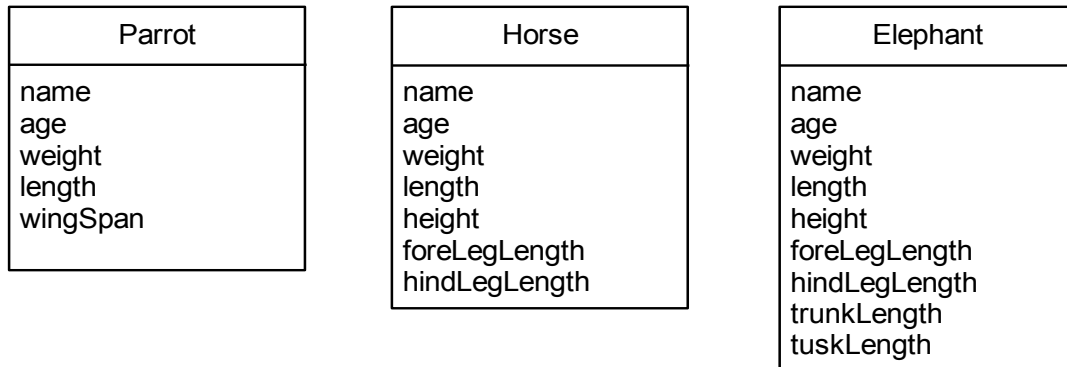
TURN OVER

Question 5

Grady Booch defined an object to be a “thing that has behaviour, state and identity”. Explain what the terms *thing*, *behaviour*, *state* and *identity* mean in this context.

Question 6

The following UML class diagram shows three classes. Use inheritance to refactor the three classes so that each attribute is defined in exactly one place. Represent the refactored structure using a UML class diagram.



Question 7

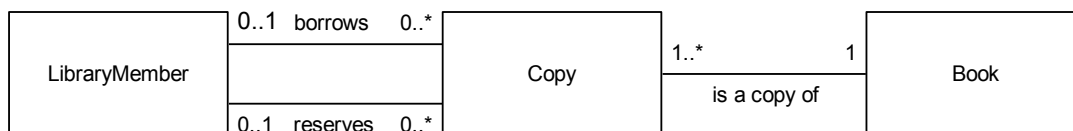
Name and briefly describe the three modes of using UML.

Question 8

What is the *noun identification technique* and what is it used for? Briefly state the steps involved in applying it.

Question 9

Study the following UML class diagram and answer the questions that follow it.

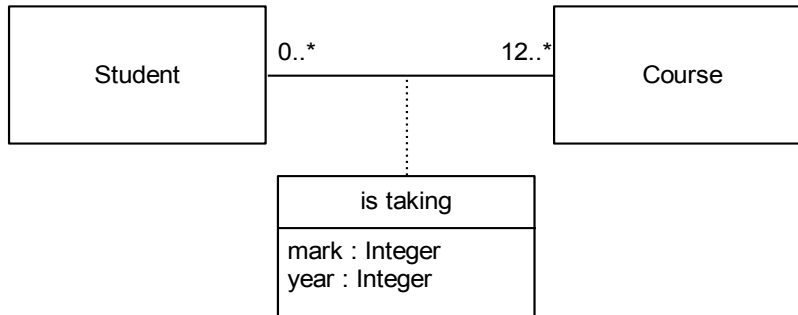


- How many Copy objects are there associated with one Book object?
- According to the diagram, is it possible for the same Copy object to be both borrowed and reserved simultaneously?
- How many Books can each library member borrow? Explain your answer.
- How many LibraryMember objects are there associated with each Copy object?

TURN OVER

Question 10

Study the following UML class diagram and answer the questions that follow it.



- What does the box labelled “is taking” represent?
- If a student takes a course twice, how can the two results that the student obtains be represented using this class model?
- Draw a diagram that represents another class model for representing the results that students obtain on courses, which uses only normal associations between classes. Does this model have any advantages over the model drawn above?

Question 11

Write down the output of the following Java program.

```
public class Integers {
    static class Ints {
        int value = 0;
    }

    public static void main(String[] args) {
        int x, y;
        Ints p = new Ints(), q = new Ints();
        x = 3; y = x; x = 4;
        System.out.println(y);
        p.value = 3; q = p; p.value = 4;
        System.out.println(q.value);
    }
}
```

TURN OVER

Question 12

Write down the output of the following Java program.

```
public class Inheritance {
    static class Point {
        int x, y;
        public String toString() {
            return "("+x+", "+y+")";
        }
    }

    static class ThreeDPoint extends Point {
        int z;
        public String toString() {
            return "("+x+", "+y+", "+z+")";
        }
    }

    public static void main(String[] args) {
        Point p = new Point();
        Point t = new ThreeDPoint();
        System.out.println(p);
        System.out.println(t);
    }
}
```

Question 13

Study the following Java program.

```
1 public class Q13 {
2     public int add(int x, int y) {
3         return x + y;
4     }
5
6     public static void main(String[] args) {
7         System.out.println(add(2,3));
8     }
9 }
```

This program will not compile because there is an error in one line. Give the line number of the line that contains the error and write down what this line should be in order for the program to generate the following output:

5

TURN OVER

Question 14

Write down the output of the following Java program.

```
import java.util.ArrayList;
import java.util.TreeSet;

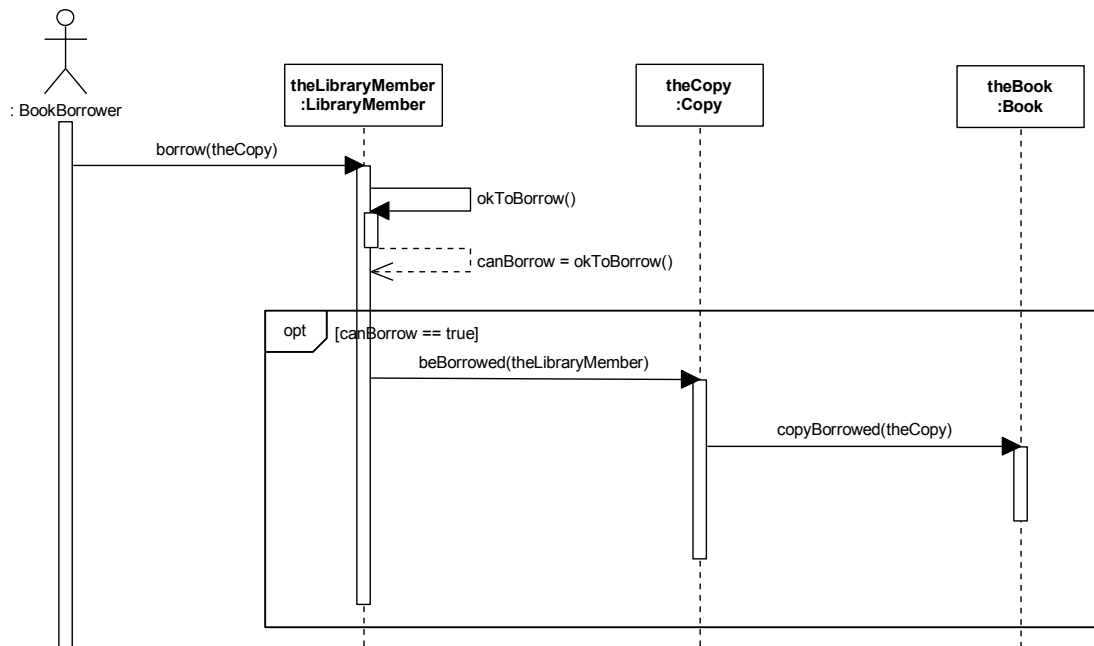
public class Collections {
    public static void main(String[] args) {
        ArrayList<Integer> a = new ArrayList<Integer>();
        TreeSet<Integer> t = new TreeSet<Integer>();

        for(int i = 0; i < 5; i++) {
            a.add(i%3);
            t.add(i%3);
        }

        System.out.println(a);
        System.out.println(t);
    }
}
```

Question 15

Study the following sequence diagram and answer the questions that follow it.

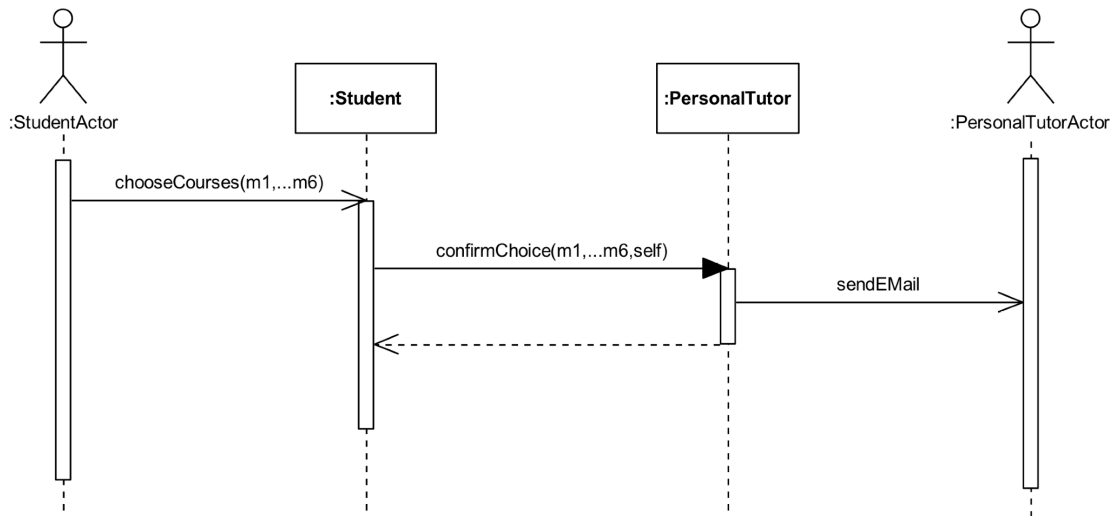


- What kind of message does the arrow labelled “borrow(theCopy)” represent?
- What does the arrow labelled “canBorrow = okToBorrow()” represent?
- What is the type of the value returned by `okToBorrow()`?
- To which class does the borrow method belong?
- To which class does the `beBorrowed` method belong?

TURN OVER

Question 16

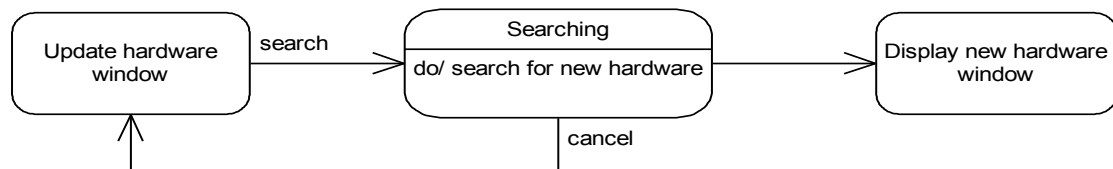
Study the following sequence diagram and answer the questions that follow it.



- What kind of thing is Student?
- What kind of message is represented by the arrow labelled sendEMail?
- To which class does the chooseCourses method belong?
- To which class does the confirmChoice method belong?
- To what kind of object does the argument “self” refer in the call to confirmChoice?

Question 17

Study the following state machine diagram, which represents the various states of a hardware-setup wizard, and answer the questions that follow it.

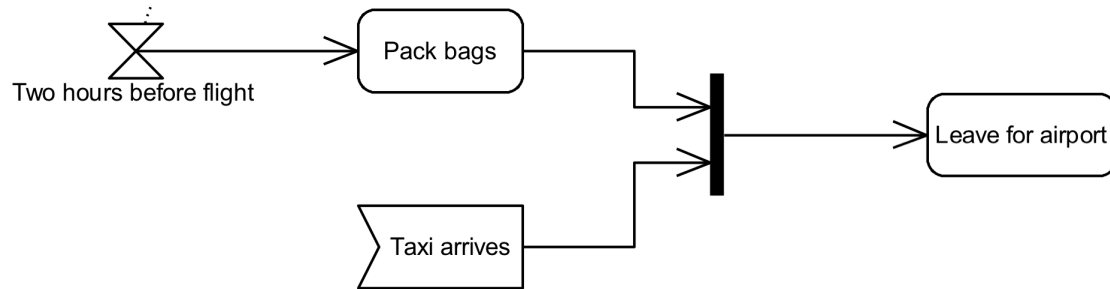


- If the system successfully finds new hardware and finishes searching for it, what happens then?
- What causes the system to transition from the “Update hardware window” state to the “Searching” state?
- Under what conditions does the system exit the “Searching” state and to which new state does it transition under each condition?
- Does “do/” indicate an action or an event?
- Are “search” and “cancel” actions or events?

TURN OVER

Question 18

Study the following UML diagram and answer the questions that follow it.



- What kind of diagram is this?
- What kind of thing is the icon labelled “Two hours before flight”? What is its function in the diagram?
- What kind of thing is the vertical bar to the left of the box labelled “Leave for airport”? How does this element function in the diagram?
- What kind of thing is the box labelled “Taxi arrives”? How does this element function in the diagram?

Question 19

Explain, with an example, what is meant by the *race condition* in multi-threaded programming.

TURN OVER

Question 20

Study the following Java program and answer the questions that follow it.

```
1  public class Threads {
2      public static Integer balance = 100;
3
4      static class ChangeRunnable implements Runnable {
5          int x;
6
7          ChangeRunnable(int x) {
8              this.x = x;
9          }
10
11         public void run() {
12             try{
13                 Thread.sleep((long)Math.random()*1000);
14                 int b = balance;
15                 Thread.sleep((long)Math.random()*1000);
16                 b += x;
17                 Thread.sleep((long)Math.random()*1000);
18                 balance = b;
19             } catch(Exception e) {
20                 e.printStackTrace();
21             }
22         }
23     }
24
25     public static void main(String[] args) {
26         try {
27             Thread plus10 = new Thread(new ChangeRunnable(10));
28             Thread minus5 = new Thread(new ChangeRunnable(-5));
29             plus10.start();
30             minus5.start();
31             plus10.join();
32             minus5.join();
33             System.out.println(balance);
34         } catch (InterruptedException e) {
35             e.printStackTrace();
36         }
37     }
38 }
```

- What are the possible outputs of this program?
- How would you modify the program so that it always generates the same output every time it is run?
- Discuss whether this modified version of the program should be considered “multi-threaded”.

TURN OVER

Question 21

Explain the difference between UDP and TCP. Which protocol would you use for each of the following applications:

- a. A time server.
- b. FTP
- c. ping

Question 22

Suppose we have two Java programs, Client and Server. The Server program is defined as follows:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(50000);
            Socket clientSocket = serverSocket.accept();
            PrintWriter out = new
                PrintWriter(clientSocket.getOutputStream(), true);
            BufferedReader in = new BufferedReader(new
                InputStreamReader(clientSocket.getInputStream()));
            String line = in.readLine();
            System.out.println(line);
            out.println(line+" to you too!");
            out.close();
            in.close();
            clientSocket.close();
            serverSocket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

TURN OVER

The Client program is defined as follows:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class Client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost",50000);
            PrintWriter out = new
                PrintWriter(socket.getOutputStream(),true);
            BufferedReader in = new BufferedReader(new
                InputStreamReader(socket.getInputStream()));
            out.println("Hello");
            System.out.println(in.readLine());
            out.close();
            in.close();
            socket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- a. When running this system, which of the two programs should be started first?
- b. What is printed out to the console window by the Server program?
- c. What is printed out to the console window by the Client program?
- d. Explain the difference between a BufferedReader, an InputStreamReader and an InputStream.
- e. To what does the number 50000 refer in the first line of the try block in each program?

Question 23

Name the three different types of thread used in a Swing program and briefly describe the function of each.

TURN OVER

Question 24

Study the following Java program and answer the questions that follow it.

```
1  import java.awt.BorderLayout;
2  import java.awt.event.ActionEvent;
3  import java.awt.event.ActionListener;
4  import javax.swing.JFrame;
5  import javax.swing.JLabel;
6  import javax.swing.JTextField;
7  import javax.swing.SwingUtilities;
8
9  public class GUI extends JFrame implements ActionListener {
10
11     private static final long serialVersionUID = 1L;
12     private JTextField text = new JTextField();
13     private JLabel label = new JLabel();
14
15     public GUI() {
16         setTitle("GUI");
17         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18         getContentPane().add(text);
19         getContentPane().add(label, BorderLayout.LINE_START);
20         text.addActionListener(this);
21         pack();
22         setVisible(true);
23     }
24
25     public void actionPerformed(ActionEvent event) {
26         label.setText("Text: "+text.getText());
27     }
28
29     public static void main(String[] args) {
30         SwingUtilities.invokeLater(new Runnable() {
31             public void run() {
32                 new GUI();
33             }
34         });
35     }
36 }
```

- What kind of layout manager does the program use? Briefly describe how this type of layout manager behaves.
- Give the variable name and type of each component that appears in the GUI.
- Which object in the program serves as an ActionListener?
- What does this ActionListener do?
- What does the statement, “pack()”, do in the sixth line of the constructor?

END OF EXAMINATION