Object Oriented Software Engineering
Problems and Exercises (1)

Graham McDonald

Outline

1. Domain Modelling
2. Control Flow Graphs

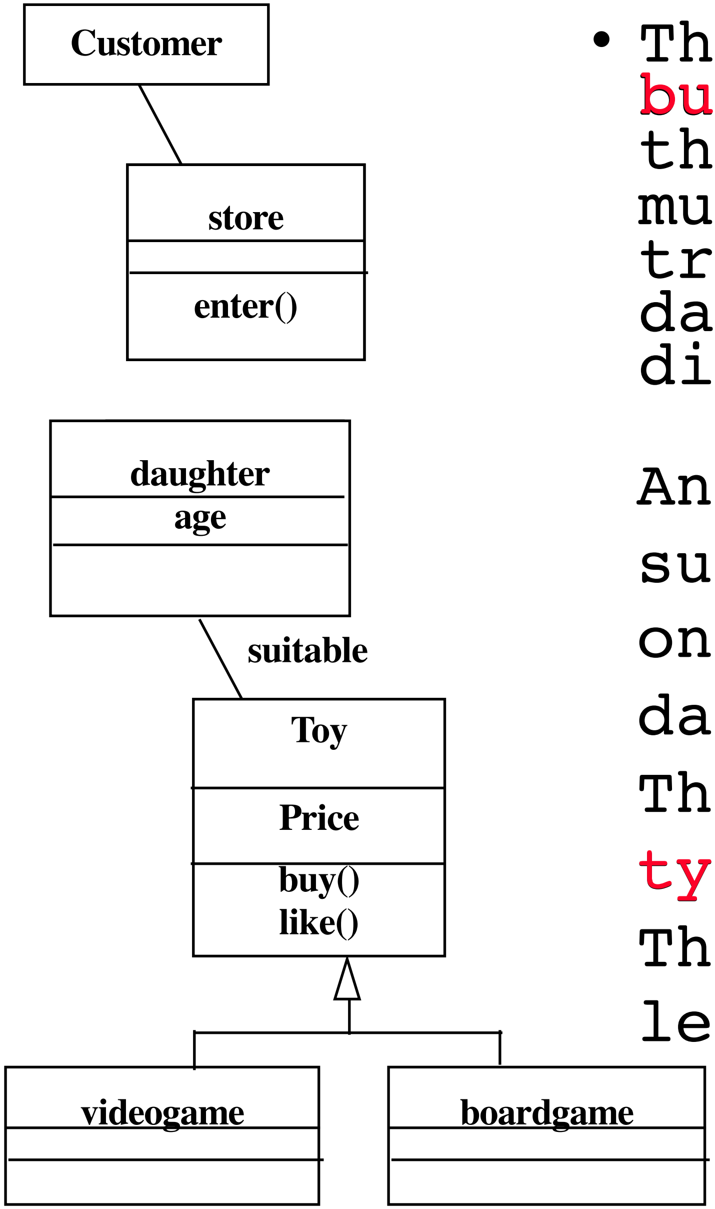
Domain Modelling

Exercise 1

Generating a Class Diagram from the following narrative:

- The customer enters the store to buy a toy. It has to be a toy that his daughter likes and it must cost less than 50 Euro. He tries a videogame, which uses a data glove and a head-mounted display. He likes it.
- An assistant helps him. The suitability of the game depends on the age of the child. His daughter is only 3 years old. The assistant recommends another type of toy, namely a boardgame. The customer buy the game and leaves the store.

Exercise 1



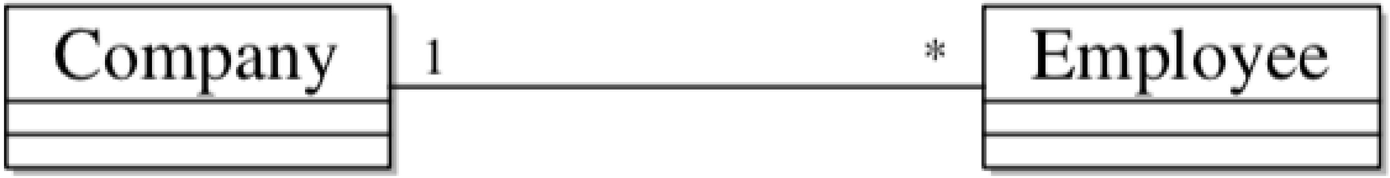
- The customer enters the store to buy a toy. It has to be a toy that his daughter likes and it must cost less than 50 Euro. He tries a videogame, which uses a data glove and a head-mounted display. He likes it.

  An assistant helps him. The suitability of the game depends on the age of the child. His daughter is only 3 years old. The assistant recommends another type of toy, namely a boardgame. The customer buy the game and leaves the store

Exercise 2

Draw the following associations using UML class diagrams:

a) A company has many employees, but the employee can only work for one company.

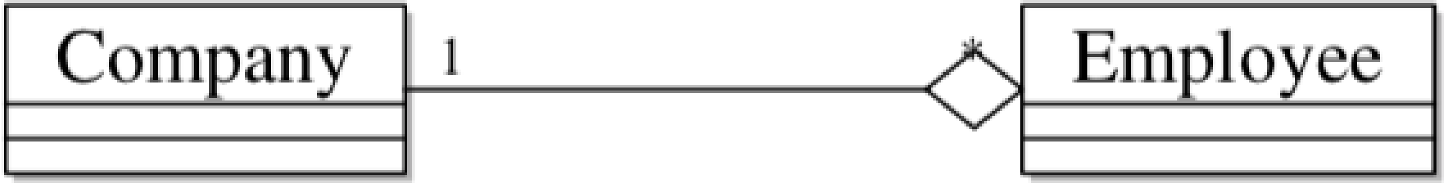


Exercise 2

Draw the following associations using UML class diagrams:

a) A company has many employees, but the employee can only work for one company.

   + Should an employee retire, then the company recruits another employee

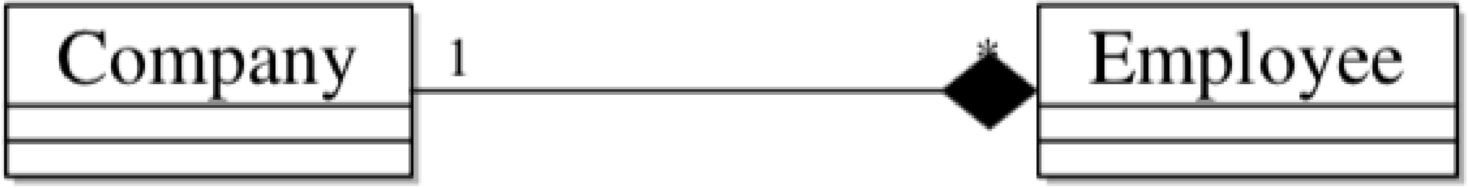


Exercise 2

Draw the following associations using UML class diagrams:

a) A company has many employees, but the employee can only work for one company.

   + Should an employee retire, then the company folds up

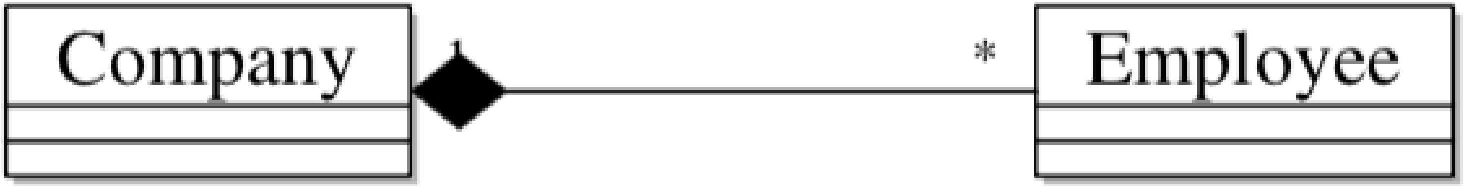


Exercise 2

Draw the following associations using UML class diagrams:

a) A company has many employees, but the employee can only work for one company.

+ Should a company be liquidated, then all employees loose their job

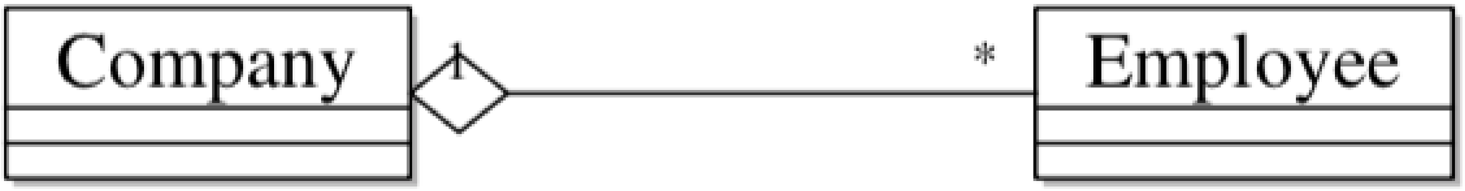


Exercise 2

Draw the following associations using UML class diagrams:

a) A company has many employees, but the employee can only work for one company.

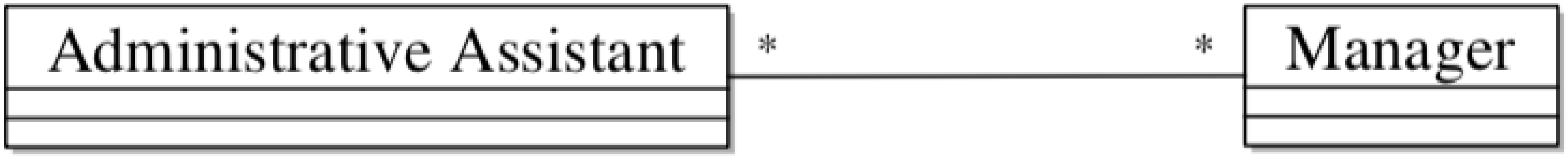
+ Should a company be liquidated, then all employees move to another company



Exercise 2

Draw the following associations using UML class diagrams:

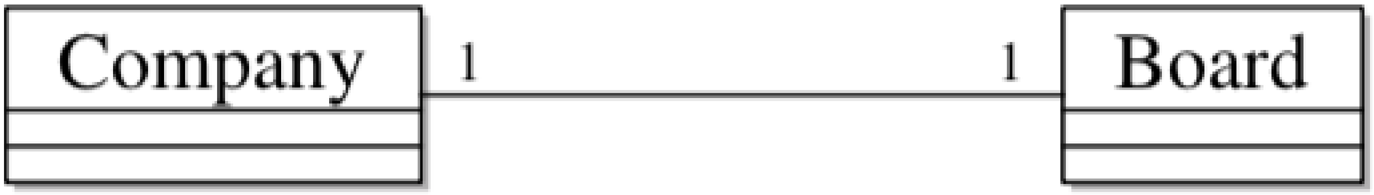
b) An administrative assistant can work for many managers, and a manager can have many administrative assistants.



Exercise 2

Draw the following associations using UML class diagrams:

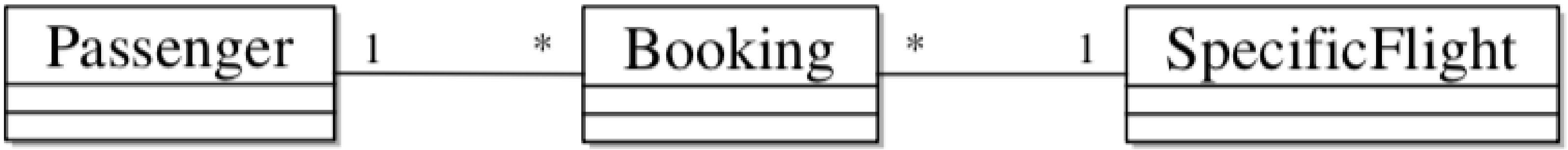
c) For each company, there is exactly one board of directors, also a board is the board of only one company.



Exercise 2

Draw the following associations using UML class diagrams:

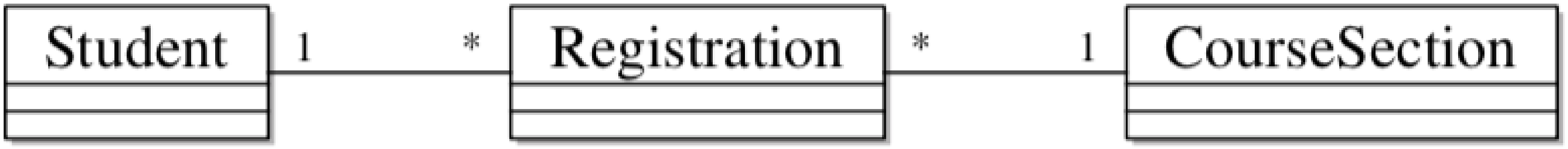
d) For each booking, there must be exactly one Passenger, but each Passenger can have any number of Bookings (e.g. on different flights and dates). Similarly, for each Booking there must always be exactly one specific flight, but each specific flight can have any number of bookings (up to the capacity of the aircraft).



Exercise 2

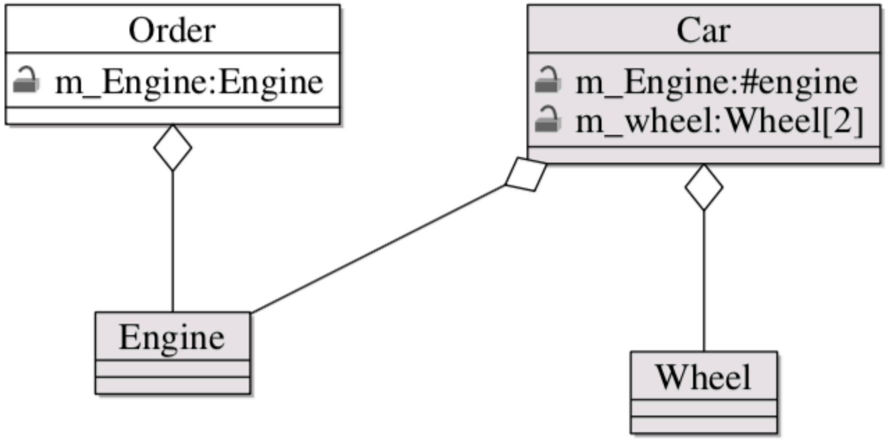
Draw the following associations using UML class diagrams:

e) A student can register in any number of course sections, and a course section can have any number of students.

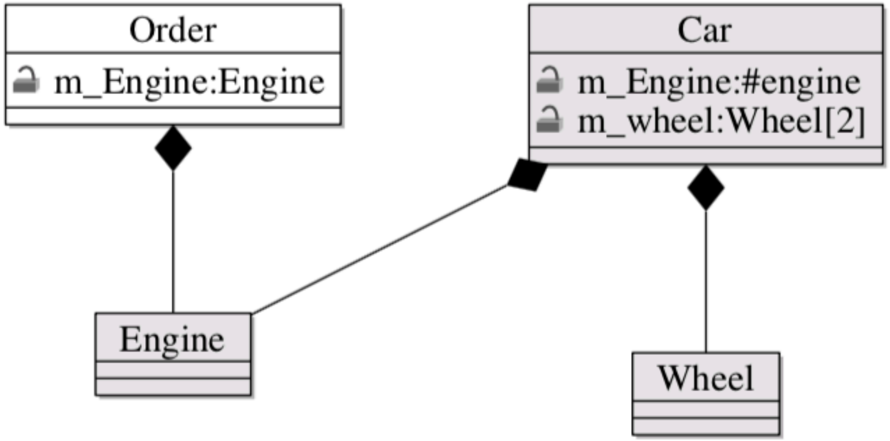


Exercise 3

- Which of the following two designs is correct:



With aggregation association

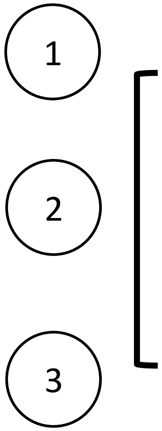


or
With composition association

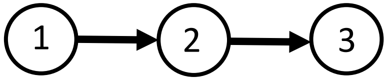
Control Flow Graphs

Exercise 1

- What is the cyclomatic complexity of the constructor below:



```
public ClassMethodCallLeak(String nodeId, String className,
                              String handlerType, String methodName,
                              String methodReturnType, String leakLine) {
        this.setNodeId(nodeId);
        this.className = className;
        this.handlerType = handlerType;
        this.methodName = methodName;
        this.setMethodReturnType(methodReturnType);
        this.setLeakLine(leakLine);
}
```

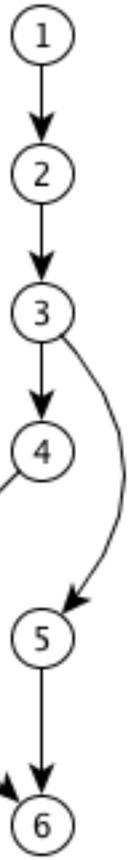


cyclomatic complexity: V(G) = 2 – 3 + 2 = 1

Exercise 2

- Draw the control flow graph for the function below:

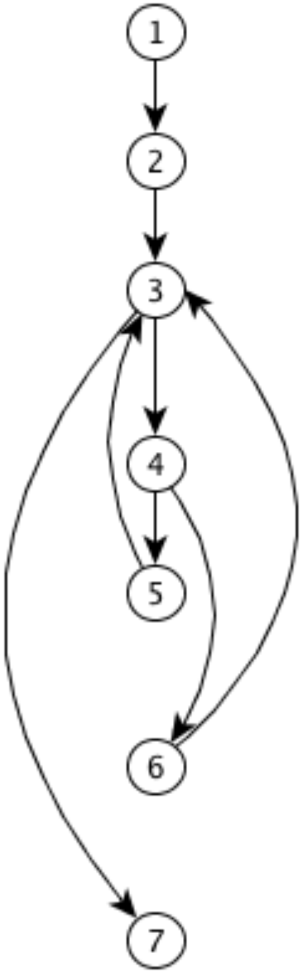
```
1     public static void main(String args[]) {
2             int a = args.length;
              int b = 3;
              int c =0;

3             if(a>b) {
4                 c =b;
              }
              else {
5                 c= a;
              }
6     }
```



Exercise 3

- Draw the control flow graph for the function below:

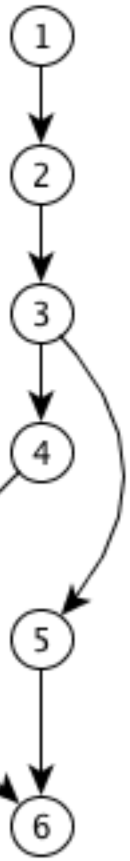
```
①   public static void main(String args[]) {
②      int a = args.length;
       int b = 3;
       int c =0;
       boolean tick = true;

③      while(tick) {
④        if(a>b) {
⑤          c =b;
         }
         else {
⑥          c= a;
           tick =false;
         }
       }
⑦   }
```



Exercise 4

- Draw the control flow graph for the outer function below:

```
1   public static void main(String args[]) {
      JButton button1 = new JButton("PressME");
      button1.addActionListener(new ActionListener() {
        @Override
2       public void actionPerformed(ActionEvent e) {
           int a = args.length;
           int b = 3;
           int c =0;
3          if(a>b) {
4            c =b;
           }
           else {
5            c= a;
           }
        }
      });
6   }
```

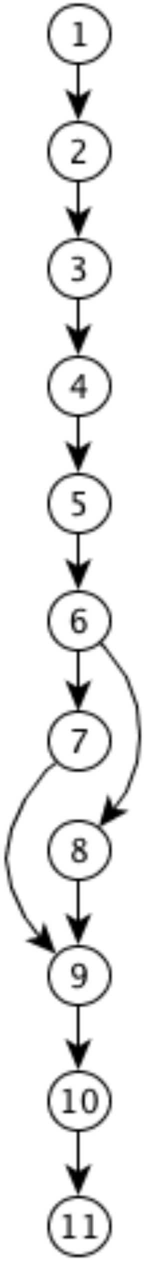


Cyclomatic complexity V(G) = 6-6+2 = 2

Exercise 5

- Draw one control flow graph for both of the functions below

```
1      public static void main(String args[]) {
2  [     JButton button1 = new JButton("PressME");
3        button1.addActionListener(new ActionListener() {
           @Override
4          public void actionPerformed(ActionEvent e) {
5  [          int a = args.length;
              int b = 3;
              int c =0;
6             if(a>b) {
7  [             c =b;
              }
              else {
8  [             c= a;
              }
9          }
10       });
11     }
```



Cyclomatic complexity V(G) = 11-11+2 = 2

Summary

- Consider different variants of domain modelling examples
- Also look at CFGs for other types of branches