Object-Oriented Analysis, Design and Programming

Medialogy, Semester 4
Monday 19 April 2010
9.00 – 12.00

You have 3 hours to complete this examination.

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

Questions 1-16 are worth 5 marks each; questions 17-23 are worth 10 marks each.

You need to get at least 50% of the total available marks in order to pass.

Question 1. (5 marks)

Which **one** of the following is **not**, in general, a benefit of encapsulation and modularity?

    A. Less for developers to learn.
    B. Easier to debug.
    C. Easier to reuse code.
    D. Makes code shorter.

Question 2. (5 marks)

Which **one** of the following best describes the cause of the Ariane 5 disaster?

    A. The project was managed by a committee consisting of members with conflicting interests.
    B. The goals of the project were too ambitious.
    C. Software was reused in a new context without adequate testing.
    D. Software interlocks did not prevent the machine from being placed in a dangerous configuration.

Question 3. (5 marks)

Which **one** of the following statements is true about the Java programming language?

    A. Each class can extend many superclasses but can only implement one interface.
    B. Each class can extend many interfaces but can only implement one superclass.
    C. Each class can implement many interfaces but can only extend one superclass.
    D. Each class can implement many superclasses but can only extend one interface.

Question 4. (5 marks)

Represent the following Java code in UML. Leave out details about how each method works internally.

```java
public class Question4 {
    private int x = 0;
    public int getX() { return x; }
    public void setX(int y) { x = y; }
}
```

Question 5. (5 marks)
Write a Java program that prints the message "Hello World!" to the standard output.

Question 6. (5 marks)

Write down the output of the following code fragment:

```java
int d = -5;
System.out.println(d + "d" + d);
System.out.println(d / 2);
System.out.println(d % 3);
```

Question 7. (5 marks)

Write down the output of the following code fragment. Marks will be given for formatting the output precisely as it would appear.

```java
for(int i = 5, j = 0; j < 5; i--, j += 2)
    System.out.println(i + " " + j);
```

Question 8. (5 marks)

Complete the following sentence by filling in the three blanks:

"An object is a *thing* that has _____ , _____ and _____ ."

Question 9. (5 marks)

The following line defines an operation in UML:

```
+ getY(x : int) : int
```

Write down (a) the *visibility* of this operation (using a complete word, not just a symbol), (b) the *signature* of this operation, (c) the *selector* of this operation, (d) the return type of this operation and (e) the argument(s) of this operation.

Question 10. (5 marks)

Suppose *a*, *b* and *c* are objects and that the following line appears in the class definition of object *a*:

b.msg(*c*)

Which **one** of the following statements is true?

    A. *a* sends the message msg to *b*.
    B. *a* sends the message msg to *c*.
    C. *b* sends the message msg to *a*.
    D. *b* sends the message msg to *c*.

Question 11. (5 marks)

What is the output of the following program?

```java
public class Question11 {
    static class Point { int x, y; }
    public static void main(String[] args) {
        Point p = new Point();
        System.out.println(p.x + " " + p.y);
    }
}
```

**Question 12. (5 marks)**

Use the noun identification technique to produce a list of candidate classes from the following requirements statement:

"*The user selects the article to be copied. The system prompts the user to provide subscriber information for the journal or to indicate a method of payment for the article. Payment can be made by credit card or by quoting an organisational account number.*"
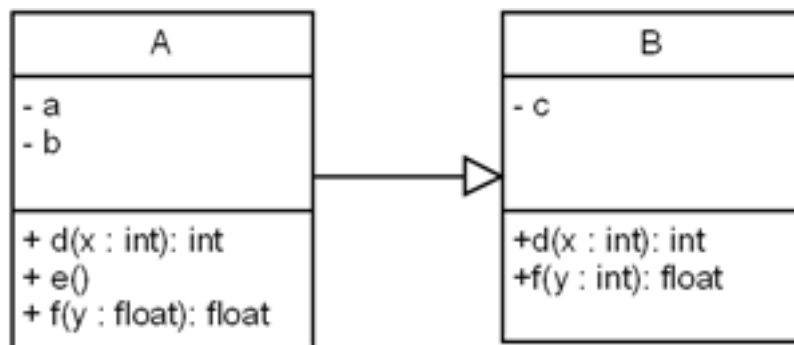
**Question 13. (5 marks)**

Study the following diagram and answer the questions below it.
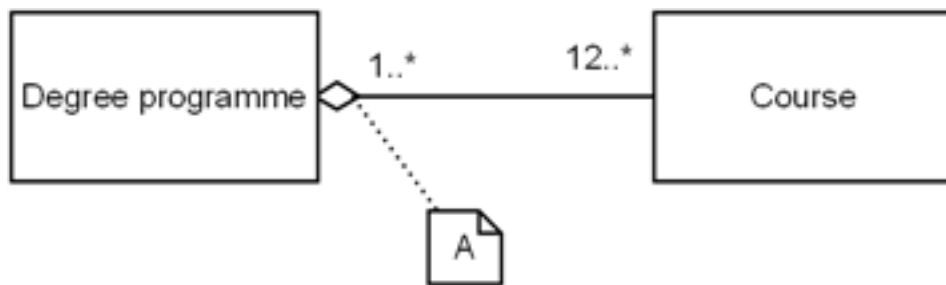


a. What kind of diagram is this?
b. What does the diamond labelled A tell us?
c. How many courses are there in each degree programme?
d. How many degree programmes can each course belong to?
e. If a degree programme is deleted, what happens to its courses?

**Question 14. (5 marks)**

Study the following diagram and answer the questions below it.



a. Is A a subclass or a superclass of B?
b. Which operations in B are overridden in A?
c. Which operations in B are overloaded in A?
d. Is variable c visible in objects of class A?

**Question 15. (5 marks)**

Which line in the following program produces an error and why? (The number at the beginning of each line indicates the line number and is not part of the code.)

```
1    public class Question15 {
2         private final double x = 1.0;
3         private final double y, z;
4         public Question15(double x, double y) {
5              this.x = x;
6              this.y = y;
7              z = 2.0;
8         }
9    }
```

continued

4

Question 16. (5 marks)

Study the following Java code.

```java
public interface A {
    int meth1(int a);
    float meth2(int b);
}

public interface B {
    double meth3();
}

public class Question16 implements A, B {
    public int XXX(int a) { return 0; }
    public double YYY() { return 0; }
    public float ZZZ(int b) { return 0; }
}
```
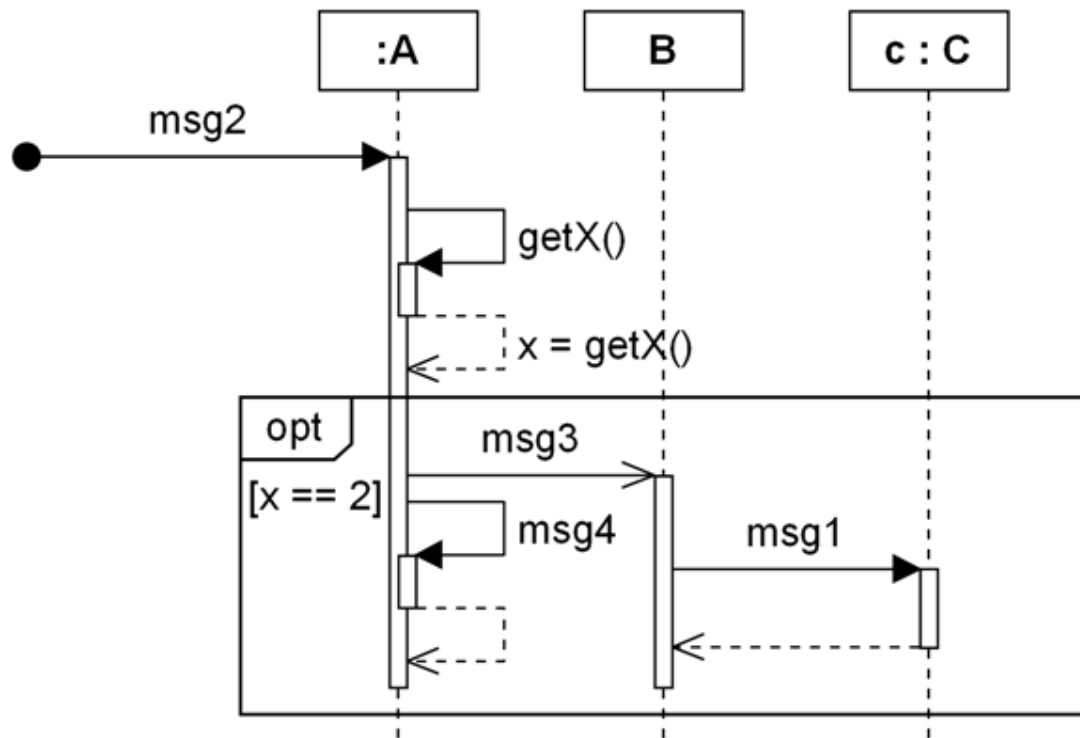
Write down the code that is needed to replace XXX, YYY and ZZZ in order to make this compile.

Question 17. (10 marks)

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

a. What kind of diagram is this?
b. What does the black disk on the left-hand end of msg2 indicate?
c. What kind of thing is msg3?
d. What kind of thing is msg1?
e. Under what condition is msg3 sent?
f. What kind of thing is B?
g. What kind of thing is c?
h. What kind of thing is A?
i. Which is sent first, msg1 or msg4?
j. Which is sent first, msg3 or msg1?

continued

Question 18. (10 marks)

Study the following diagram and answer the questions that follow it.
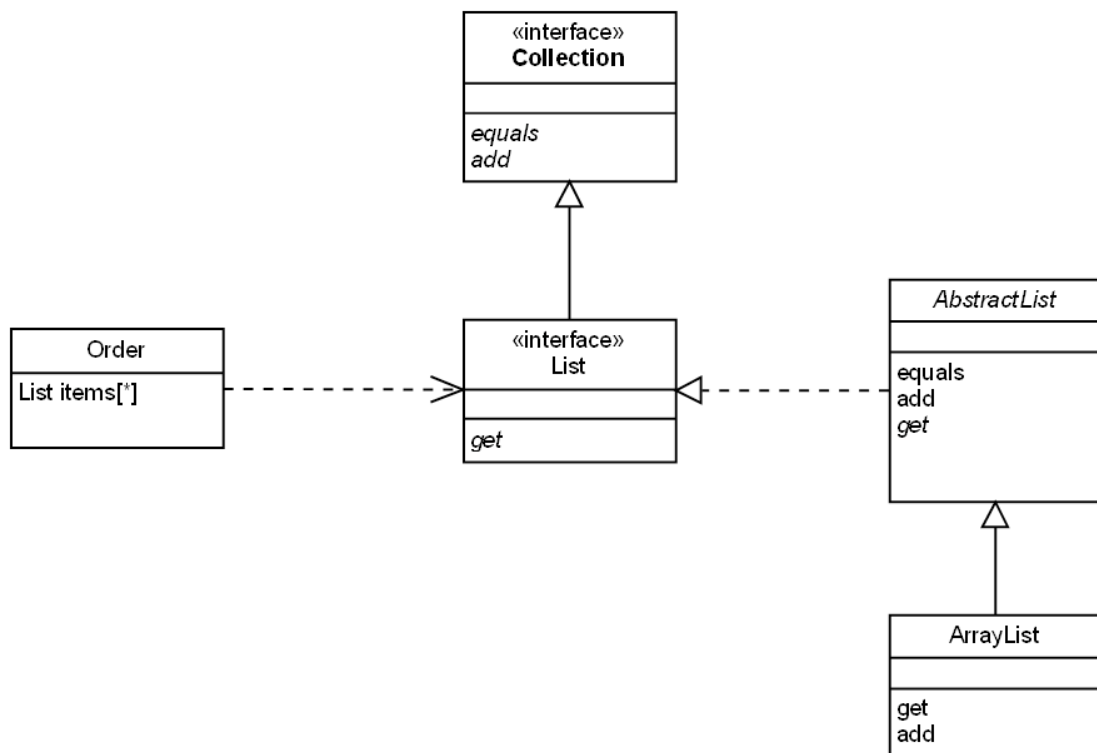
a. What kind of diagram is this?
b. Name the concrete classes in this diagram.
c. Name any abstract classes in this diagram.
d. Name the abstract methods in the diagram and indicate which interface or abstract class they belong to.
e. What does the arrow from AbstractList to List signify?
f. What does the arrow from List to Collection signify?
g. What does the arrow from Order to List signify?
h. What does the arrow from ArrayList to AbstractList signify?
i. Can AbstractList be instantiated? Explain your answer.
j. If the diagram shows all the interfaces and classes in the system, what concrete class will actually be used to implement the *items* attribute in Order?

Question 19. (10 marks)

Fill in the blanks, *AAAAAAA* to *HHHHHHH* in the following code, given that, when it is run, it generates the output:

FOO
BAR

Here is the code that you have to complete:

```java
public interface Shape {
    void display();
}


public class Circle AAAAAAA {
    public void BBBBBBB() {
        System.out.println("CCCCCCC");
    }
}


public class Square DDDDDDD {
    public void EEEEEEE() {
        System.out.println("FFFFFFF");
    }
}


public class Shapes {
    public static void main(String[] args) {
        ArrayListGGGGGGG list = new ArrayListHHHHHHH();
        Circle c = new Circle();
        Square s = new Square();
        list.add(c);
        list.add(s);
        for(Shape x : list)
            x.display();
    }
}
```
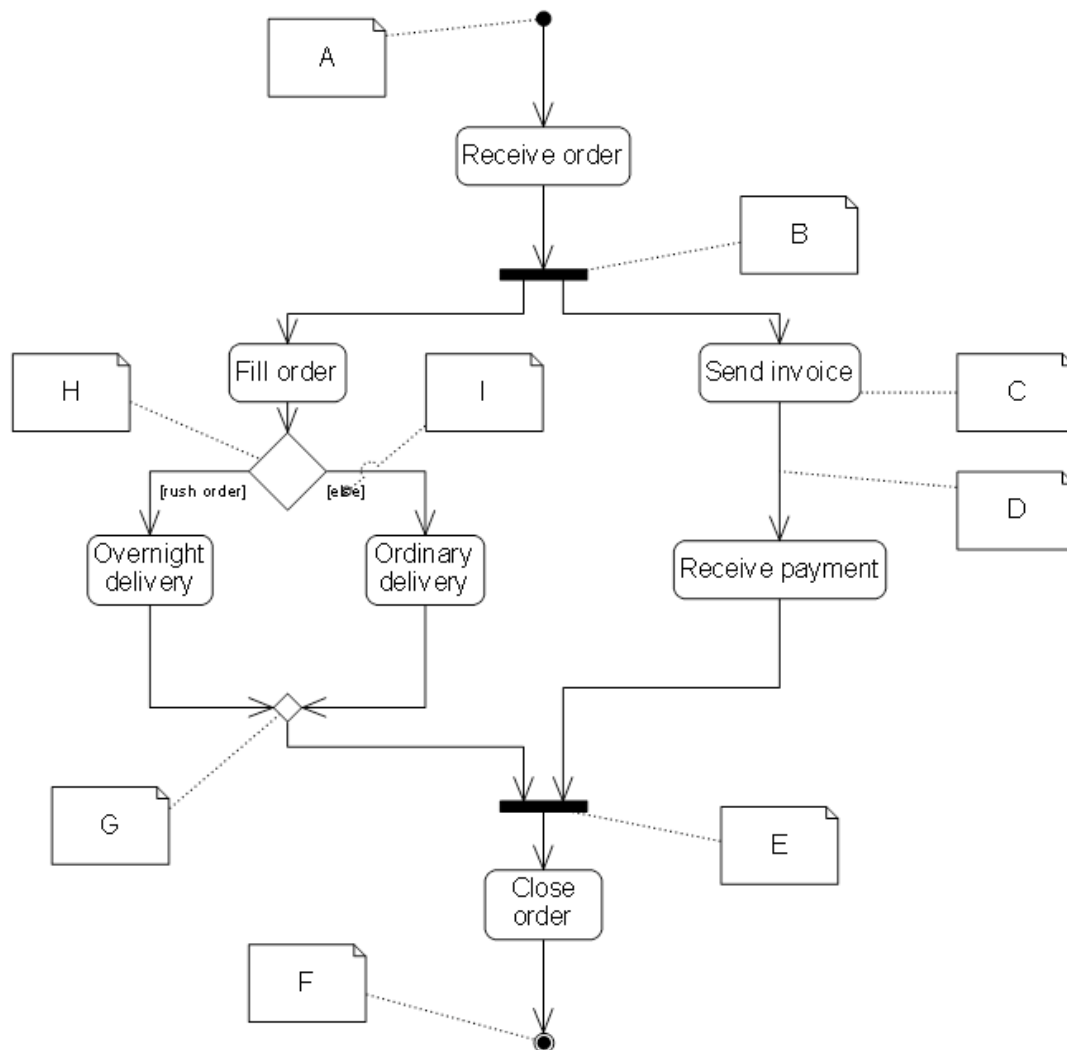
Question 20. (10 marks)

What is the output of the following program?

```java
public class Question20 {
    static class BurpException extends Exception {
        private static final long serialVersionUID = 1L;
        public BurpException(String s) {
            super(s);
        }
    }
    static class Burp {
        public Burp(String s) throws BurpException{
            if (isValidBurp(s))
                throw new BurpException(s);
            else
                throw new BurpException("Invalid");
        }
        private boolean isValidBurp(String s) {
            if (s.length() == 4 && s.charAt(0) == 'B'
                    && s.charAt(3) == 'p')
                return true;
            return false;
        }
    }
    public static void main(String[] args) {
        String[] burps = {"Barp", "Blip", "Flip", "Flop",
                            "Boop", "Stop"};
        for (int i = 0; i < 5; i++)
            try {
                new Burp(burps[i]);
            } catch (BurpException b) {
                System.out.println(b.getMessage());
            }
    }
}
```

9

Question 21. (10 marks)

Study the following diagram and answer the questions that follow it

a. What sort of diagram is this?

b. Write down what each of the labels A-I should say.

Question 22. (10 marks)

Draw a state machine diagram for the interface of a simplified cash machine that behaves as follows:

The interface starts by showing a Welcome screen. When the user inserts a bank card, a Login screen is presented. While the login screen is showing, the user can either enter a PIN, or press Cancel. If the user enters a correct PIN, an Amount screen is presented that allows the user to specify the amount of money required. If an incorrect PIN is entered and there have been fewer than three attempts, the machine shows an Incorrect PIN page. If three incorrect PINs have been entered, the machine returns to the Welcome page and does not return the user's card. If the user presses Cancel on the Login screen, the machine returns to the Welcome screen and the card is returned. While the Amount screen is showing, the user can either press Cancel or enter an amount of money. If the user presses Cancel on the Amount screen, the machine returns to the Welcome screen and the card is returned. If an amount of money is entered on the Amount screen and this amount of money is available to the user, then this amount of money is dispensed, the machine returns to the Welcome screen and the card is returned to the user. If an unavailable amount of money is entered on the Amount screen, the machine returns to the Welcome screen and returns the card to the user.

Question 23. (10 marks)

Study the following code and draw a UML class diagram to describe it. Assume all the code occurs within the same package. Remember that in UML you usually don't specify the internal implementational details of methods.

```
public interface A {
    int a(int x);
    float b(float y);
}
public class B implements A {
    private int c;
    private float d;
    public int a(int x) { return 2 * x; }
    public float b(float y) { return 2 * y; }
    public int getC() { return c; }
    public float getD() { return d; }

}
public class C extends B{
    public int a(int x) { return 3 * x; }
}
public class D {
    private C e;
    public C getE() { return e; }
}
```

END OF EXAM