Name: Number:

Object Oriented Programming 2014/15

Final Exam
June 19, 2015

Directions (read carefully):

- CLEARLY print your name and ID on every page.
- The exam contains 8 pages divided into 4 parts. Make sure you have a complete exam.
- The point value of each problem is indicated next to the problem and below.
- You have two hours.
- It is wise to skim all the problems before beginning, to best plan your time.
- This is a closed book exam. No notes of any kind are allowed. Do all work in the space provided, using the backs of sheets if necessary.
- **Over the table it should ONLY be this exam, a pen and an ID.**
- Turn off the mobile phone. The use of a mobile phone annuls the exam.

Part	Problem	Description	Page	Marks
I	1 a) b)	UML	2	1.5
II	2 a) b) c) d) e)	Java development	3	3.0
	3 a) b) c) d) e)	Java multiple choice	6	2.5
	4	Java miscellaneous	7	1.0
III	5	XML	8	1.0
IV	6	Swing	8	1.0
Total				**10.0**

Name:	Number:

Part I -- UML (1.5 marks)

1 – Consider a movie shop where users can buy and rent movies. The title, the year, and the movie's director identify a movie. A movie-shop user may or not be a movie-shop subscriber. A user contains a name and an ID. A subscriber is a user that also contains a mobile number and an address. Only subscribers can hire a movie; users, subscribers or not, can always buy a movie. When a movie is not available to sell, it is ordered. In this case, the movie shop maintains a list of orders of movies to buy; an order may be related to more than one movie to buy. The user and the ordering date identify the order. The list of bought and hired movies should be stored in the respective user and/or subscriber. A subscriber has its own card from the movie shop. The card has a code and points. The movie-shop maintains a list of the issued cards.

a) [1.0 mark] Define the UML class diagram for the presented problem.

Name: Number:

b) [0.5 marks] Define an UML object diagram considering a subscriber with one rent movie and two bought movies. Set all needed attributes/associations with some dummy values.

Part II -- Java (6.5 marks)

2 – Consider an interface ICard with two no-arg methods, getSuit and getRank, both returning a String. The ICard makes also available a list of suit strings ("spades", "hearts", "diamonds", "clubs") and rank strings ("ace", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", "jack", "queen", "king"). Provide an implementation of this interface in a class named Card. Then, provide a Deck that contains a list of cards (with all suits and ranks). The Iterator and ArrayList are the only types you can use from java.util package. Note: Iterable if from java.lang package.

a) **[0.5 marks]** Provide the ICard interface.
b) **[0.5 marks]** Provide the Card class where the corresponding suit and rank should be constant and visible only inside the class. Its only constructor receives two strings, one identifying the rank and the other the suit. In the Card class provide a textual description of the respective card as "R of S", where R is the rank and S is the suit.
c) **[0.75 marks]** Provide the Deck class implementing the Iterable interface (for that consider a DeckIterator). The no-arg Deck constructor should build a deck and store the cards in an ArrayList; in the end shuffle the cards using the static method: void java.util.Collections.shuffle(List<?> list).
d) **[0.75 marks]** Provide the DeckIterator implementation. The hasNext method returns true iff there are more cards in the deck. The next method returns the next card in the deck. The remove method should throw an UnsupportedOperationException exception.
e) **[0.5 marks]** Provide a main method in a Main class where a Deck is built and iterated through a for-each loop printing the cards to the terminal.

Name: Number:

```
public class Exam1 {
      public interface ICard {

            String getSuit();
            String getRank();

            List<String> suits = Arrays.asList("spades", "hearts",
                        "diamonds", "clubs");
            List<String> ranks  = Arrays.asList(
                        "ace", "two", "tree", "four",
                        "five", "six", "seven", "eight",
                        "nine", "ten", "jack", "queen", "king");
      }

      public static class Card implements ICard{

            private final String rank;
            private final String suit;

            public Card(String rank, String suit){
                  this.rank = rank;
                  this.suit = suit;
            }

            @Override
            public String getSuit() {
                  return suit;
            }
            @Override
            public String getRank() {
                  return rank;
            }

            @Override
            public String toString() {
                  return rank + " of " + suit;
            }
      }
```

Name:	Number:

```
	public static class Deck implements Iterable<ICard> {

		private ArrayList<ICard> cards = new ArrayList<ICard>();

		public Deck(){
			for(String s : ICard.suits){
				for(String r : ICard.ranks){
					cards.add(new Card(r, s));
				}
			}
			Collections.shuffle(cards);
		}

		@Override
		public Iterator<ICard> iterator() {
			return new DeckIterator();
		}

		public class DeckIterator implements Iterator<ICard> {

			Iterator<ICard> listIter = cards.iterator();

			@Override
			public boolean hasNext() {
				return listIter.hasNext();
			}

			@Override
			public ICard next() {
				return listIter.next();
			}

			@Override
			public void remove() {
				throw new UnsupportedOperationException();
			}
		}
	}

	public static void main(String[] args) {

		Deck d = new Deck();

		for(ICard c : d){
			System.out.println(c);
		}
	}
}
```

Name:	Number:

3 – Fill the answers of multiple-choice questions in the following table (use only capital letters). If you want to correct your answer scratch out and write the correct answer. Each correct question is marked 0.5 points. A question not answered is marked 0 points, whereas a wrong answer discounts 0.2 points. If you think NONE of the options are correct, write NONE.

Question	a)	b)	c)	d)	e)
Answer	A	C	C	A	A

a) **[0.5 marks]** Which of the following does not display 9.95:

A. System.out.println(9.+95/100)
B. System.out.println(995/100.0)
C. System.out.println(9+0.95)
D. System.out.println(9.+95.0/100)
E. System.out.println(9+"."+95)

b) **[0.5 marks]** When MyClass is compiled the java compiler gives an error message: MyClass is not abstract and does not override abstract method **<some method>** in java.lang.Comparable. Which of the following could replace **<some method>** in the error message?

A. equals(Object obj)
B. compare(MyClass mc1, MyClass mc2)
C. compareTo(MyClass mc)
D. compareTo(Object obj)
E. compare(Object obj1, Object obj2)

c) **[0.5 marks]** What is the size of an int variable in Java?

A. 2 bytes
B. 8 bytes
C. 4 bytes
D. It depends on the compiler settings
E. It depends on the operating system

d) **[0.5 marks]** Let lli1 and lli2 be both a LinkedList<Integer> with values {0,1,2} and {3,4,5}, respectively. What is the output from the following code?

```
Iterator<Integer> it1 = lli1.iterator();
Iterator<Integer> it2 = lli2.iterator();
it1.next();
it1.next();
it2.next();
it1.remove();
it2.remove();
System.out.print(lli1);
System.out.println(lli2);
```

A. [0,2][4,5]
B. [0,1,2][3,4,5]
C. [0,1][3,5]
D. [2][4,5]
E. The code does not compile

Name: Number:

e) **[0.5 marks]** What is the value of `x` after the following statements are executed?

A. `-1`
B. `1`
C. `-0.5`
D. `-0.75`
E. `0`

```
int x = 1;
x += f(g(x)) - g(f(x));
```

```
public static int f(int x) { return x + 2; }
public static int g(int x) { return x * 2; }
```

4 – [1.0 marks] Explain why polymorphism promotes code to be extensible. Provide an example of that.

Name:	Number:

Part III -- XML (1 mark)

5 – [1.0 marks] Consider the problem presented in Part I and, particularly, problem 1b), that is, a subscriber with one rent movie and two bought movies. Informally, present an XML document, with XML elements and attributes, to store all information needed.

Part IV -- Swing (1 mark)

6 – [1.0 marks] Explain what a model-view-controller (CMV) architecture consists in.