Name: Number:

Object Oriented Programming 2008/09

Final Exam
July 10th, 2009

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)	Java development	4	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	C++	8	1.0
Total				**10.0**

Name: Number:

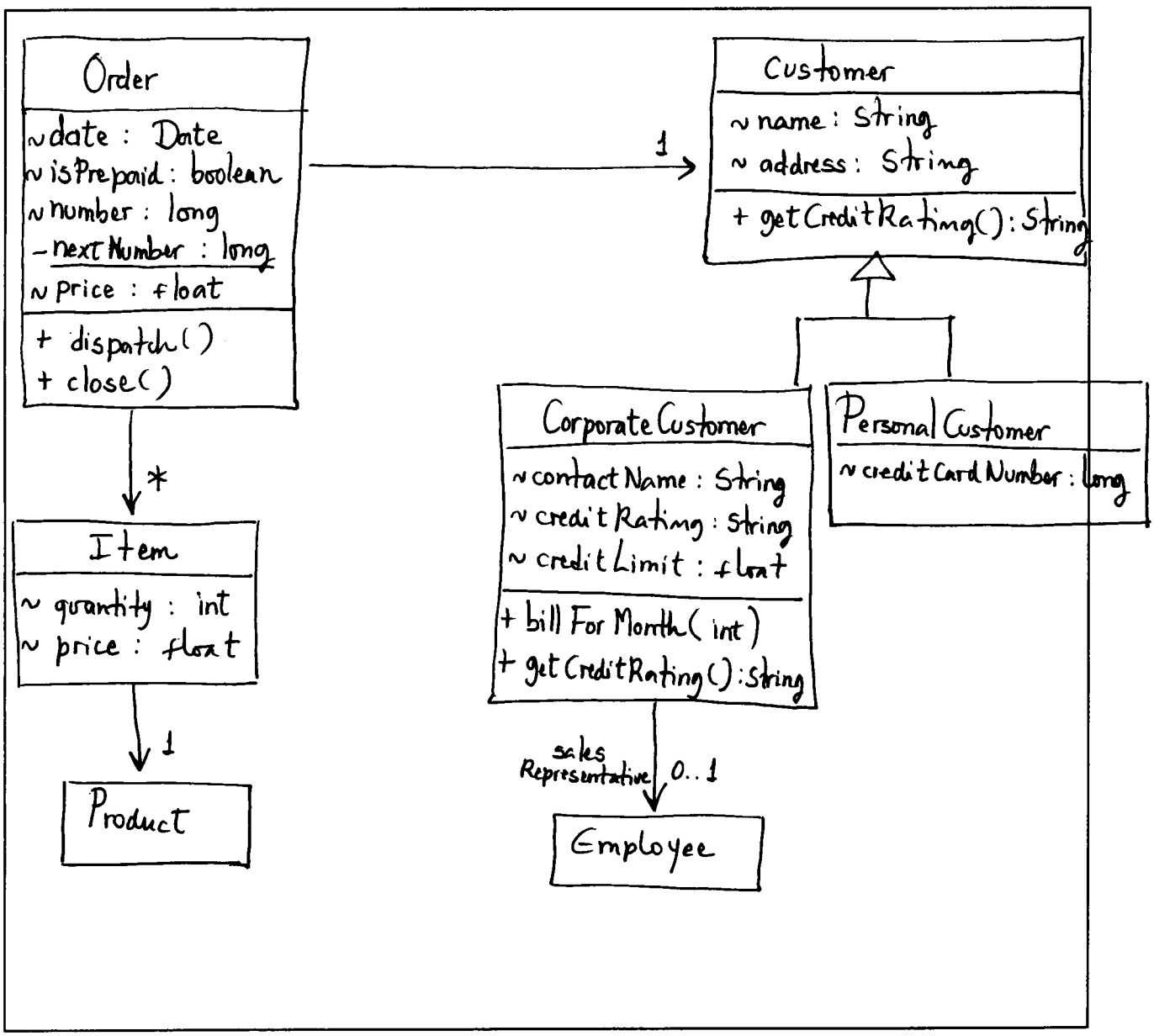
Part I -- UML (1.5 marks)

1 – Consider that an on-line shop intends to develop an order processing application. An order should have: (i) the date in which it was received; (ii) if it is prepaid; (iii) a sequential number that must be set automatically whenever an order is made; and (iv) the total price of the order. Orders should be dispatched to the customer and closed after that.

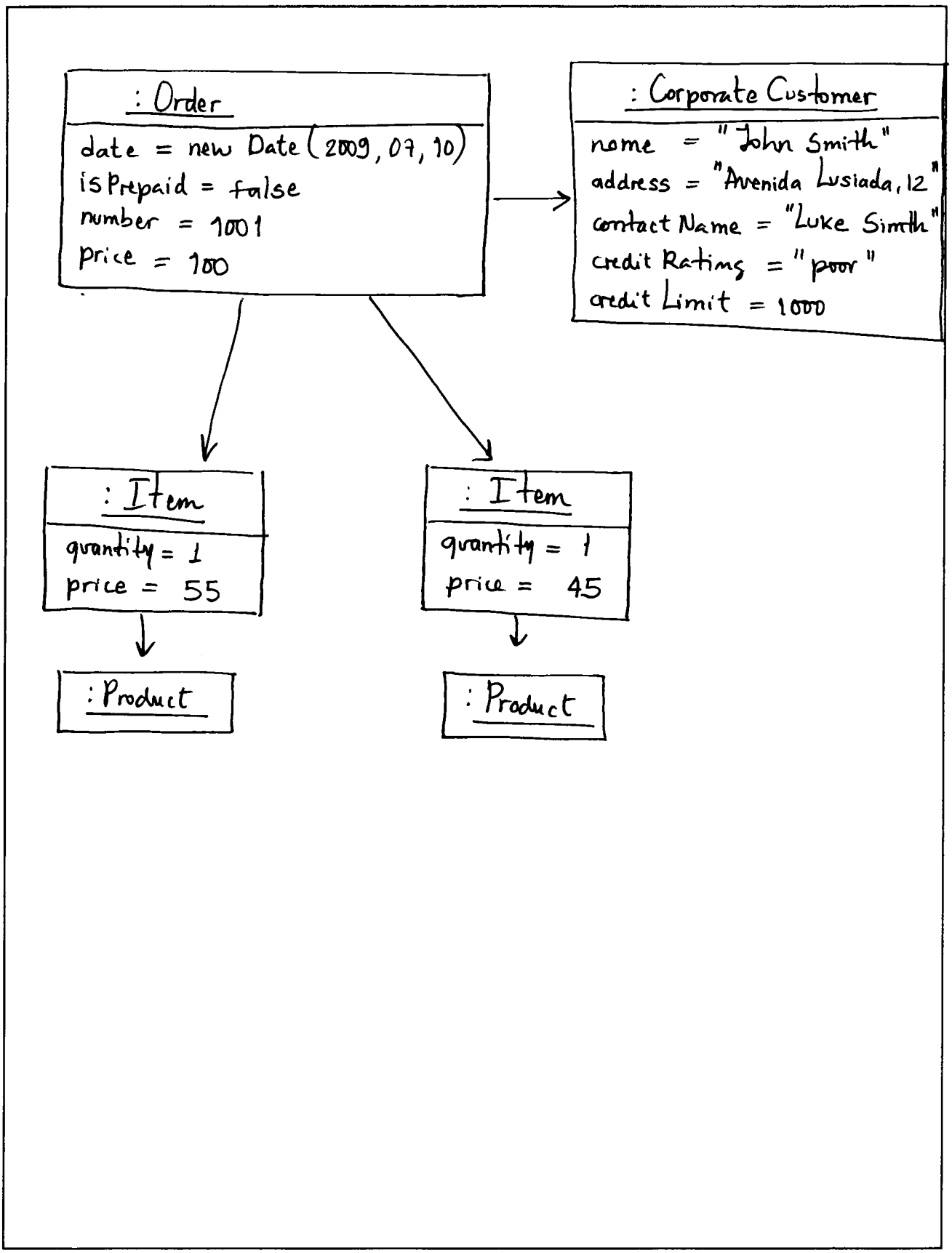
An order may contain different items, each one with a different product. Each item should have the product, the quantity being ordered for that product, and the unitary price of the product.

A customer stores information about its name and address. A customer may be a corporate customer or a personal customer. A corporate customer should store information about its contact name, credit rating ("poor" or "rich") and credit limit. A corporate customer must be associated with one sales representative, which is an employee of the on-line shop. The system should be able to create a bill that will be sent to these customers every month. A personal customer should store its credit card number and its credit rating is always "poor". It should be possible to get the credit rating polymorphically from a generic costumer with a getter method.

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



b) [0.5 marks] Define an UML object diagram considering that a corporate customer has ordered two different products from the on-line shop. Assume that the sequential number of the respective order is 1001. Set all other attributes with some dummy values.



Name: Number:

Parte II -- Java (6.5 marks)

2 – Consider a class Ingredient that encapsulates the name of the ingredient and its quantity in grams. We intend to develop such a class, providing a constructor which receives the attributes needed to properly initialize an object. Define getters (selectors) for both attributes and a setter (modifier) only for grams attribute. Ingredient attributes should be accessed outside the class only by such methods.

a) **[0.75 marks]** Define the `Ingredient` class.

```
public class Ingredient {
      private String name;
      private int grams;

      public Ingredient(String name, int grams) {
            this.name = name;
            this.grams = grams;
      }
      public String getName() {
            return name;
      }
      public int getGrams() {
            return grams;
      }
      public void setGrams(int grams) {
            this.grams = grams;
      }
}
```

b) **[0.5 marks]** Implement the `toString` method to present a description of an ingredient like: `"Apple (112 grams)"`.

```
@Override
public String toString() {
      return name + " (" + grams + " grams)";
}
```

c) **[0.75 marks]** Implement the `equals` method to return true if two objects are deeply the same (the same name and quantity).

```
@Override
public boolean equals(Object obj) {

      // Standard equals() tests...
      if (this == obj) return true;
      if (!(obj instanceof Ingredient)) return false;

      // Now do deep compare
      Ingredient other = (Ingredient)obj;
      return (grams==other.getGrams() && name.equals(other.getName()));
}
```

d) **[1.0 marks]** Define a `main` method creating 3 ingredients: an apple `a` with 112 grams and two bananas `b1` and `b2` both with 236 grams. Print the following information to the terminal:

- Their description (of `a`, `b1` and `b2`)
- Their equality (between `a` and `b1`, `a` and `b2`, and `b1` and `b2`)
- Their equivalence (between `a` and `b1`, `a` and `b2`, and `b1` and `b2`)

commenting after every printing instruction the printed result.

```
public static void main(String args[]) {

      Ingredient a = new Ingredient("Apple", 112);
      Ingredient b1 = new Ingredient("Banana", 236);
      Ingredient b2 = new Ingredient("Banana", 236);

      //Implicit call of toString to print their description
      System.out.println(a);   // Apple(112 grams)
      System.out.println(b1); // Banana (236 grams)
      System.out.println(b2); // Banana (236 grams)

      //Equality
      System.out.println(a==b1);  // false
      System.out.println(a==b2);  // false
      System.out.println(b1==b2);  // false

      //Equivalence
      System.out.println(a.equals(b1));  // false
      System.out.println(a.equals(b2));  // false
      System.out.println(b1.equals(b2));  // true

}
```

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.

Question	a)	b)	c)	d)	e)
Answer	D	C	D	B	B

a) **[0.5 marks]** What is the value of the static attribute `x` after a second object of type `X` being created?

```
class X {
      static { x+=2; }
      static int x=-1;
      public X() { x*=3; }
}
```

A. `3`
B. `18`
C. `-9`
D. `9`
E. None of the above

b) **[0.5 marks]** Assume that we are importing `java.util.Arrays`. What is it printed to the terminal?

```
int[] tiVar1 = new int[3];
int[] tiVar2 = tiVar1;
Arrays.fill(tiVar2,10);
tiVar1[1]=5;
System.out.print(Arrays.toString(tiVar1));
System.out.print(Arrays.toString(tiVar2));
```

A. `[0,5,0][10,5,10]`
B. `[0,5,0][10,10,10]`
C. `[10,5,10][10,5,10]`
D. `[10,5,10][10,10,10]`
E. None of the above

c) **[0.5 marks]** What is it printed to the terminal?

```
class X{
      int xpto(){return 5;}
}
class Y extends X{
      int xpto(){return 10;}
      void test(){
            X x = (X) this;
            System.out.print(this.xpto());
            System.out.print(x.xpto());
            System.out.print(((X)this).xpto());
            System.out.print(super.xpto());
      }
      public static void main(String[] args){
            new Y().test();
      }
}
```

A. `5 5 5 10`
B. `10 5 5 10`
C. `10 10 5 5`
D. `10 10 10 5`
E. None of the above

d) **[0.5 marks]** What is it printed to the terminal when the method `foo` is called?

```
public static void foo(){
      try {
            System.out.print(1);
            return;
      } catch (Throwable e) {
            System.out.print(2);
      } finally {
            System.out.print(3);
      }
}
```

A. `12`
B. `13`
C. `123`
D. `1`
E. None of the above

e) **[0.5 marks]** Which syntax would you use to express a wildcard with a lower bound of some `type`?

A. `?` B. `? super type` C. `? extends type` D. None of the previous

4 – [1.0 marks] Singleton classes represent objects for which only one instance should exist. Implement in Java a singleton class called Universe.

```
public final class Universe {

      public static Universe getInstance() {
            return uINSTANCE;
      }

      //Single instance created upon class loading
      private static final Universe uINSTANCE = new Universe();

      //private constructor prevents construction outside the class
      private Universe() {}

}
```

Parte III -- XML (1 mark)

5 – [1.0 marks] Consider the problem presented in Part I and, particularly, problem 1b), that is, consider a corporate costumer has ordered two different products. Informally, present a XML document, with XML elements and attributes, to store all information at hand.

```
<order number="1001" prepaid="false">
      <date>10/07/2009</date>
      <totprice>100</totprice>
      <items>
            <item>
                  <quantity>1</quantity>
                  <unitprice>55</unitprice>
                  <product>mouse</product>
            </item>
            <item>
                  <quantity>1</quantity>
                  <unitprice>45</unitprice>
                  <product>keyboard</product>
            </item>
      </items>
      <customer>
            <name>John Simth</name>
            <address>Anedida Lusiada, 12</address>
            <corporate creditrating="poor" creditlimit=1000>
                  <contactname>Luke Simth</contactname>
            </corporate>
      </customer>
</order>
```

Parte IV -- C++ (1 mark)

6 – [1.0 marks] Explain what is a reference in C++. Give a simple example of reference usage.

```
A reference in C++ is just an alternative name to the same object. In the
following example, i and r are just two alternative names to the same int.

int i=1;
int& r=1; // r and i refers to the same int
int x=r;  // x=1
r=2;      // i=2
```