

STUDENT NAME: _____

STUDENT ID: _____

SECOND MIDTERM

COMP-250: Introduction to Computer Science - Winter 2008

March 14, 2008

You are allowed one double sided cheat sheet.

There are 4 questions, for a total of 100 points. Please read all the questions first. Please make sure to **write your name** and ID number on the exam booklet!

Answer all questions on the exam booklet

Good luck!

1. [15 points] **Visibility rules and initialization in Java**

For each of the following pieces of Java code, explain what will happen when you try to execute it:

(a)

```
public class TestMyClass {
    public static void main(String[] args) {
        int x = 2;
        System.out.println(x);
    }
}
```

Prints 2.

(b)

```
public class TestMyClass {
    public static void main(String[] args) {
        int x;
        System.out.println(x);
    }
}
```

There will be a compilation error because the variable x has not been initialized

(c)

```
public class TestMyClass {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++)
            System.out.println('*');
        System.out.println(i);
    }
}
```

There will be a compilation error because variable i is declared only inside the for loop, and the last printing statement, which attempts to print i, is outside the for loop. So the compiler complains that the variable is not defined.

2. [15 points] Objects and classes

Consider the following small class:

```
public class MyClass {
    private int a;
    private int b;
    MyClass() {
        a = 1;
        b = 2;
    }
    public int get_a() { return a; }
    private int get_b() { return b; }
}
```

For each of the following pieces of code, explain what happens when you compile it and, if appropriate, when you execute it.

(a)

```
public class TestMyClass {
    public static void main(String[] args) {
        MyClass x;
        System.out.println(x.get_a());
    }
}
```

There will be an error because x has to be constructed before any methods can be called on it.

(b)

```
public class TestMyClass {
    public static void main(String[] args) {
        MyClass x = new MyClass();
        System.out.println(x.get_a());
    }
}
```

Prints 1

(c)

```
public class TestMyClass {
    public static void main(String[] args) {
        MyClass x = new MyClass();
        System.out.println(x.get_b());
    }
}
```

There will be a compilation error because we are trying to call a private method. Private methods can only be called inside other methods of the class.

3. [60 points] **Inheritance, parameter passing**

Consider the two following classes:

```
public class MyParentClass {
    int x;
    public MyParentClass() { x=1; }
    public int get_x() { return x; }
    public void change_x() {
        x=x+5;
    }
    public void set_x(int val) { x=val; }
}

public class MyChildClass extends MyParentClass {
    public MyChildClass() { super(); }
    public void change_x() {
        x=x+10;
    }
}
```

(a) What will be the result of the following main function:

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

    MyParentClass obj = new MyParentClass();
    System.out.println(obj.get_x());
    obj.change_x();
    System.out.println(obj.get_x());

    MyChildClass c = new MyChildClass();
    System.out.println(c.get_x());
    c.change_x();
    System.out.println(c.get_x());

}
```

Prints the following:

```
1
6
1
11
```

(b) Now suppose that you change MyChildClass as follows:

```
public void change_x() {
    super.change_x();
    x=x+10;
}
```

How does the output of the main method change?

The last number printed will be 16 instead of 11.

(c) Now suppose that you are working with the testing class below. Describe what happens

```
public class MyTestClass {

    public static void my_method(MyParentClass obj) {
        obj.set_x(20);
    }
    public static void another_method(int a) {
        a = 10;
    }
    public static void main (String[] args) {

        MyParentClass o = new MyParentClass();
        System.out.println(o.get_x());
        my_method(o);
        System.out.println(o.get_x());

        MyChildClass c = new MyChildClass();
        System.out.println(c.get_x());
        my_method(c);
        System.out.println(c.get_x());

        int a = 50;
        System.out.println(a);
        another_method(a);
        System.out.println(a);
    }
}
```

Prints the following:

```
1
20
1
20
50
50
```

4. [10 points] **Code design**

This is an open-ended question. Suppose you are asked to design a system for scheduling courses at McGill. As part of it, you need to take care of describing different types of classes (e.g. lecture-based, lecture and lab, multi-term). You also need to be able to describe constraints (e.g. prerequisites). Describe what kind of classes and structure you might use.

For this question, many solutions were possible. The basic ideas I was looking for were:

- Define a “course” general class, which has a title, a list of instructors, a list of TAs, a list of prerequisites.
- Define sub-classes for the different types of courses, and each sub-class will have specific information. E.g. A lecture class will have a room, a starting time, a duration, and a day slot (MWF or TuTh). A lecture and lab class will also have lab times and rooms, and may have other restrictions.
- A general container (e.g. a list) will be used to maintain the collection of all the course objects.
- The scheduling system would be a separate class, which uses the lecture objects.