

Please fill in your Student Number and, optionally, Name.

Student Number : \_\_\_\_\_

Name : \_\_\_\_\_

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2007

## Supplementary Theory Test 3A Solution

Question	Mark	Max	Initials
1		10	
2		10	
3			
<b>TOTAL</b>		<b>30</b>	

**Marks : 30**

**Time : 40 minutes**

**Instructions:**

- a) Answer all questions.
- b) Write your answers in the space provided.
- c) Show all calculations where applicable.

## Question 1: OOP Concepts. [10]

a) What is overloading? [2]

*multiple methods with the same name/return value [1] but different parameter lists [1]*

b) Briefly discuss one advantage of overloading. [1]

*there are no longer multiple names for methods that essentially do the same thing [1]*

c) What is encapsulation? Why is it important? [2]

*data and methods in the same class [1] it makes it easier for programmers to associate data with methods = fewer errors/faster programmer/less effort [1]*

d) Explain the purpose of each of the 3 modifiers typically used for constant declarations. [3]

*public = make it accessible from outside [1]*

*static = make it always available even without an instance [1]*

*final = cannot be changed [1]*

e) Why can a static method not call a non-static method? [2]

*static methods are not associated with any particular instance [1] so Java cannot determine which instance the method is invoked on. [1]*

## Question 2: Class Definitions [10]

Consider the following class definition and answer the questions that follow.

```
class Complex
{
    private double real;
    private double imaginary;

    public Complex ( double r, double i )
    {
        real = r;
        imaginary = i;
    }
    public Complex ( double r )
    {
        real = r;
        imaginary = 0;
    }

    public double getReal ()
    {
        return real;
    }

    public String toString ()
    {
        if (Math.abs (imaginary) > 0)
        {
            if (imaginary < 0)

                return "" + real + imaginary + "i";
            else
                return real + "+" + imaginary + "i";
        }
        else
            return "" + real;
```

```
}  
}
```

- a) Write a statement to create a variable of this type and assign to it an object corresponding to the real number 42, using the most efficient overloaded constructor. [2]

```
Complex c [1] = new Complex (42); [1]
```

- b) Write a mutator for the instance variable named **imaginary**. [3]

```
public void setImaginary [1] ( double i ) [1]  
{  
    imaginary = i; [1]  
}
```

- c) Explain exactly what the output of this **toString** method is. [2]

*if the number is complex, return in the format a+bi or a-bi [1]*

*if the number is real, return only a real number as a string [1]*

- d) Write a method to add another **Complex** object to the current one. Assume the existence of appropriate accessors. [3]

```
public add ( Complex x ) [1]  
{  
    real += x.getReal(); [1]  
    imaginary += x.getImaginary(); [1]  
}
```