

University of Bath

DEPARTMENT OF COMPUTER SCIENCE
EXAMINATION

CM10227: PROGRAMMING 1

Tuesday, 26 January 2010, 16.30–18.30

No calculators may be brought in or used.

Full marks will be given for correct answers to THREE questions. If you opt to answer more than the specified number of questions, you should clearly identify which of your answers you wish to have marked. In cases where you have failed to identify the correct number of answers the marker is only obliged to consider the answers in the order they appear up to the number of answers required.

1. (a) Explain the concept of polymorphism. [3]
- (b) Why is data encapsulation important? [3]
- (c) Explain the different ways parameters can be passed. How is this done in Java and how is it done in Python? [5]
- (d) Consider the following (not really useful but error-free) program::

```
def main(op):
    d={}
    def f1(i):
        seq[i+1]=seq[i+1]-seq[i]

    def f2(i):
        i=i+1
        seq[i]=seq[i]*i

    def f3(i):
        j = i + 1
        seq[j]=seq[j]-seq[i]
    d[2]=f3

    d={1:f1, 2:f2, 3:f2}
    seq=[6,5,1,10,3,4,8,2,7]
    i=0
    j=0
    while i<len(seq)-1:
        di = j%(len(d)-1)
        if j<len(op):
            d[op[di]](i)
        else:
            j=0
            d[op[di]](i)
        i = i + 1
        j = j + 1

    print seq
```

Question 1 continues on next page ...

Question 1 continued ...

We ran this piece of code and we obtained the following output:

```
[6, 5, -4, 14, -11, 15, -7, 9, -2]
```

What actual parameter did we pass to the function main? You might want to include the method you used to obtain the answer. That way, if you made a mistake in your answer but the your reasoning was correct you will not loose all the marks. [9]

2. (a) What are abstract data types and why are they useful? [3]
 (b) What is meant by class level and object level methods/variables? How is the difference expressed in Java? [3]
 (c) In Python, if a list is passed as an argument to a function, any changes to the list parameter will have an effect on the original argument. If the same is done with tuples, this will not be the case. Why is this? How is this difference represented in memory? [5]
 (d) Predict the output of the following (not really useful but error-free) program:

```
class ClassA {
    static String name = "ClassA";
    int number = 1;
    String m[] = {"A", "B", "C", "D", "E",
                  "F", "G", "H", "I"};

    String getM() {
        return m[number];
    }

    public String toString() {
        return name + ": " + getM();
    }

    ClassA(int number) {
        this.number = number;
        System.out.println("ClassA created");
    }

    ClassA() { System.out.println("ClassA created");}
}

class ClassB extends ClassA {
    static String name = "ClassB";

    ClassB(int n) {
        super(n-1);
        System.out.println("ClassB created");
    }

    public String toString() {
        return name + ": " + getM();
    }
}
```

```

    }

    class ClassC extends ClassA {
        static String name = "ClassC";

        ClassC(int n) {
            super(n-1);
            System.out.println(" ClassC  created");
        }
    }

    public class Weird {
        public static void main(String[] args) {
            ClassA[] cbs = new ClassA[9];
            int i = 0;
            while(i<2)
                cbs[i++] = new ClassA();
            while(++i<7)
                cbs[i] = ((i++>=3)?new ClassC(i):new ClassA());
            while(i<9)
                cbs[i] = new ClassB(i++%2==0?i<8?4:5);
            for(int k=0; k<cbs.length; k++)
                System.out.println(""+cbs[k]);
        }
    }

```

You might want to include the method you used to obtain the answer. That way, if you made a mistake in your answer but the your reasoning was correct you will not loose all the marks. [9]

3. (a) Explain the concept of scope. What is its relation with name overloading? [3]
- (b) Explain the workings of the call-stack in Python. [4]
- (c) How does bubblesort work? What is the complexity of this algorithm? [5]
- (d) Write a recursive and iterative function/method that implements the behaviour of the modulo operator for two positive integers. You are not allowed to use the one provided by the language nor are you allowed to use division. One version should be implemented in Python, the other one is Java. [8]

4. (a) Give two reasons why documentation is important? [2]
(b) Explain the concepts of abstraction and modularisation. [4]
(c) Explain the similarities and differences of Python and Java when it comes to types. [4]
(d) Implement mergesort in either Python or Java. [10]

5. (a) When deciding on which algorithm to use for a certain task, e.g. sorting, what aspects should be taking into account? [3]
(b) Explain recursion and iteration. What do they have in common and how do they differ? [5]

- (c) Consider the following Java code:

```
if ((a-->4) || (b++<a))
{
    System.out.println("true");
    System.out.println("a: "+ a);
    System.out.println("b: "+ b);
}
else
{
    System.out.println("false");
    System.out.println("a: "+ a);
    System.out.println("b: "+ b);
}
```

Rewrite this code using

- (i) Two if statements
(ii) A single if statement without the use of either || or | [4]
(d) Implement a priority queue in Java. [8]