

University of Bath

**DEPARTMENT OF COMPUTER SCIENCE
EXAMINATION**

CM10227: PROGRAMMING 1

Thursday, 24 January 2013, 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.

8

1. (a) Write a recursive **and** an iterative version of exponentiation (power, e.g a^x). One version needs to be implemented in Java and the other one in Python. The function/method should be able to operate with any positive integer as an exponent. If a negative is given, the result should be -1.
You are **not** allowed to the built-in operator/method/function for exponentiation. [10]
- (b) What is garbage collection? [2]
- (c) What are mutable data types in Python? What effect do they have when passed as parameters? [4]
- (d) Explain aliasing in the context of variables. Provide a Java example of variables that are aliases and provide an example where two variables are referring to the same value but are not aliases. [4]

2. (a) Consider the following piece of code:

```
# this function takes a list of numbers and three functions
# as arguments.
def wrongOutput(numbers, func1, func2, func3):
    # It will apply an inner function f to each element
    # f returns the application of a function on its parameter.
    # The function applies func1 if the element is divisible by three
    # If not, it applies func2 if the number is even
    # or func3 if the number is odd and larger than 8
    # as a default it returns -1
    def f(x):
        if (x / 3 == 0):
            return func1(x)
        if (x / 2 == 0):
            return func2(x)

    for el in numbers:
        el = f(el)
```

While the comments are correct, something went wrong with the code. When we run:

```
# correct code
def func1(x): return 100
def func2(x): return x/2
def func3(x): return x*3
numbers1 = [1,2,3,4,5,6,7,8,9]
print numbers1
wrongOutput(numbers1, func1, func2, func3)
print numbers1
```

we expected to see:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, -1, 100, 12, 15, 100, 21, 24, 100]
```

Instead we got:

```
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
```

Remove the logical error(s) and explain what went wrong. [10]

- (b) We discussed two ways of differentiating programming languages on the basis of data types. Explain both and mention in which categories you place Python and Java [4]
- (c) What is the difference between syntax and run-time errors (bugs)? Are there other kinds of bugs? [3]
- (d) Explain the concept of polymorphism. [3]

3. (a) Predict the output of the following (not really useful but error-free) program:

```

class ClassA {
    int number = 2;
    String m[] = {"A", "B", "C", "D", "E", "F", "G", "H", "I"};
    String getM() {
        return(m[number]);
    }
    public String toString() {
        return getM();
    }
    ClassA(int number) {
        this.number = number;
    }
    ClassA() { }
}

class ClassB extends ClassA {
    ClassB(int n) {
        super(n+1);
    }
    public String toString() {
        return super.toString();
    }
}

public class Weird {
    public static void main(String[] args) {
        ClassA[] cbs = new ClassA[9];
        int i = 0;
        while(i<3)
            cbs[i++] = new ClassA();
        while(i<6)
            cbs[i++] = i>=5?new ClassB(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.print(""+cbs[k]);
    }
}

```

[7]

- (b) While the above code compiles and runs it suffers from bad code structure. Identify three ways on how the code's structure could be improved. [3]
- (c) What are the visibility modifiers in Java? Explain each of them briefly. [3]
- (d) Implement a class that cannot be subclassed yet instantiated without the use of the keyword "final". Demonstrate the instance creation of your class [7]

4. (a) In Python, implement an ADT for a bank account. You should be able to request the balance, deposit, withdraw money and ask for a print-out of the last 5 transactions. On creation, name and pin should be provided, and if desired a balance
- Make sure your code is robust (e.g. no possibility of withdrawing negative amounts or going overdrawn) and that only a valid pin is accepted. At any given time you are only allowed to store 5 transactions.
- Demonstrate the usage of your ADT. [10]
- (b) Why is data encapsulation important? [3]
- (c) Explain the concept of a constructor in the context of Java classes. [3]
- (d) What are object and class diagrams? [4]
5. (a) Design a Java program which reflects the roles and the hierarchy of university members.
- For each of the following roles, create a class/interface/abstract class (as appropriate): university member, student, PhD student, investigator, academic, lecturer, professor.
- For this, assume that academics as well as PhD students are investigators and that PhD student is a subclass of student. Lecturers and professors are academics.
- Everyone is a university member.
- Add to the appropriate classes data fields for name, subject (field of study), research topic (for investigators) and courses taught by an academic.
- Add methods for getting and setting the content of these fields. An implementation of the classes is not required. [9]
- (b) Explain how the call stack in Python works. [4]
- (c) In Java, given two boolean expressions A and B, is the following statement true or false? Briefly explain your answer.
- $A \mid B$ is the same as $B \mid A$ for any Boolean expressions A and B.
- [3]
- (d) In the context of Java, what is meant by the static **and** dynamic type of a single variable. Provide an example of when they are different. In the example, identify the static and the dynamic type. [4]