

University of Bath

**DEPARTMENT OF COMPUTER SCIENCE
EXAMINATION**

CM10227: PROGRAMMING 1

Tuesday, 17 January 2012, 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 a constructor in the context of Java classes. [3]
 - (b) In Java, given two boolean expressions A and B, is the following statement true or false? Briefly explain your answer.
$$A \text{ — } B \text{ is the same as } B \text{ — } A \text{ for any Boolean expressions A and B.}$$

[3]
 - (c) What are mutable data types in Python? What effect do they have when passed as parameters? [4]
 - (d) 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]
2.
 - (a) Why is code robustness considered desirable for a program? Give one example how this could be achieved. [2]
 - (b) In the context of Java, what is meant by the static and dynamic type of a variable. Provide an example of when they are different. In the example, identify the static and the dynamic type. [5]
 - (c) 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]

Question 2 continues on next page . . .

Question 2 continued ...

- (d) 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 f
    # f returns the application of a function on its parameter.
    # func1 if the element can be divided by 3.
    # If not, it applies func2 if the number is even
    # or func3 if the number is odd.
    def f(x):
        if (x / 3 == 0):
            return func1(x)
        if (x / 2 == 0):
            return func2(x)

    for el in numbers:
        el = f(el)

def func1(x): return 0
def func2(x): return x/2
def func3(x): return x*3
```

While the comments are correct, something went wrong with the code. When we ran the following code

```
numbers1 = [1,2,3,4,5,6,7,8]
print numbers1
wrongOutput(numbers1, func1, func2, func3)
print numbers1
```

we expected to see:

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

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.

[9]

3. (a) Explain the concepts of method overloading and overriding [4]
(b) 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]
(c) Explain the concepts of abstraction and modularisation. [4]
(d) A deque ("double-ended queue") is like a queue, except that it allows access at both ends. Create a Java class Deque which implements a deque. It should implement the following interface:

```
public interface DequeInterface {  
    public void addFront(Object o);  
        // adds item o to the front of the deque  
  
    public void addRear(Object o);  
        // adds item o to the rear of the deque  
  
    public Object removeFront();  
        // removes the item at the front and returns it  
  
    public Object removeRear();  
        // removes the item at the rear and returns it  
  
    public int size();  
        // returns the number of items which are currently  
        // stored in the deque  
}
```

[8]

4. (a) Name and explain the three types of errors/bugs one can have in a program. [3]
(b) How does method look-up work in Java? Explain both compile and run-time. [5]
(c) In Java, the keyword `abstract` indicates that a class cannot be instantiated. However, the class type can still be used as a static type. How can the same behaviour be achieved without the use of the keyword `abstract`. Provide a code example of the class and demonstrate its use as a static type. [8]
(d) In Java, how are the values of variables stored. [4]

5. (a) What are pure functions and how are they related to functional programming? [3]
(b) Explain exception handling in Python [4]
(c) Explain the concept of incremental programming [3]
(d) Write a recursive **and** an iterative version of modulo. One version needs to be implemented in Java and the other one in Python. The function/method should be able to operate with positive and negative numbers.
You are **not** allowed to use built-in operators for division and modulo (i.e. the use of the '`\`' or '`%`' operator is not permitted). [10]

