

The University of Nottingham

SCHOOL OF COMPUTER SCIENCE AND INFORMATION
TECHNOLOGY

A LEVEL 1 MODULE, SPRING SEMESTER 2002-2003

INTRODUCTION TO PROGRAMMING

Time allowed TWO Hours

Candidates must NOT start writing their answers until told to do so

Answer FOUR out of SIX questions

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a dictionary to translate between that language and English provided that neither language is the subject of this examination. No electronic devices capable of storing and retrieving text may be used.

DO NOT turn your examination paper over until instructed to do so.

Additional Material:

1. Description of the Hashmap class

1. (a) Write an implementation of the method

```
public static void printStars(int n) (5)
```

which prints a line of *n* stars * to the screen. Here is the result of `printStars(5)`:

```
*****
```

- (b) Write an implementation of the method

```
public static void printTriangle(int n) (5)
```

which prints a triangle of *n* levels to the screen. The first level has one star, and each next level has one more star than the previous one. Here is a triangle of 5 levels:

```
*  
**  
***  
****  
*****
```

- (c) Write an implementation of the method

```
public static void printTree(int n) (10)
```

which prints a tree of *n* levels to the screen. The first level has one star, and each next level has two more stars than the previous one. Here is a tree of 5 levels:

```
  *  
 ***  
*****  
*****  
*****
```

- (d) Suppose that the `printTree()` method is a member of the `AsciiDrawing` class. How would you call this method from the `main()` of some other class to print a tree with 13 levels? (5)

2. (a) What are wrapper classes in Java and what are they used for? (5)
(b) Write an implementation of a wrapper class `Byte` which has the following constructor: (10)

```
public Byte (byte b)
```

which creates a wrapper object for the byte `b`, and a method

```
public byte byteValue()
```

which returns the byte value of the wrapper.

- (c) Using your `Byte` class, write a class called `PeriodicTable` which stores pairs (chemical element description as a `String`, atomic number as a `byte`), for example, ("hydrogen", 1) or ("calcium", 20) in a hashtable (use `java.util.HashMap` class, API description is enclosed). The first element of the pair is the key and the second element the value. (9)

The `PeriodicTable` class should have methods

```
public void store(String description, byte number)
```

which puts the pair in the hashtable, and

```
public byte retrieve(String description)  
throws NoSuchElementException
```

which returns the number stored with this description; if there is no entry corresponding to this description, the method should throw an exception. You may assume that the exception class is implemented as follows:

```
public class NoSuchElementException extends Exception {  
  
    public NoSuchElementException(String s) {  
        super("There is no chemical element called " + s);  
    }  
}
```

- (d) The largest atomic number is 118. What is the maximal value of a `byte` type variable? (1)

3. Consider the following program:

```
1 class Pair {
2     private static int first;
3     private static int second;
4
5     Pair(int i, int j){
6         first = i;
7         second = j;
8     }
9
10    public void display(){
11        System.out.println("(" + first + ", " + second + ")");
12    }
13
14    public static void main(String[] args){
15        int[] source = new int[8];
16        source[0] = 0;
17        for (int i=1; i < source.length; i++){
18            source[i] = ++source[i-1];
19        }
20        for (int i=0; i < source.length; i++){
21            System.out.println(source[i]);
22        }
23        Pair[] destination = new Pair[4];
24        for (int i=0; i < destination.length; i++){
25            destination[i]= new Pair(source[2*i], source[2*i+1]);
26            destination[i].display();
27        }
28        for (int i=0; i < destination.length; i++){
29            destination[i].display();
30    } }
```

(a) What will be printed after the loop on lines 20-22 is executed? (5)

Question continued overleaf

- (b) What will be printed after the loop on lines 24-27 is executed? (5)
- (c) What will be printed after the loop on lines 28-30 is executed? (5)
- (d) What is the difference between static and instance fields? (5)
- (e) What would change in the program behaviour if `first` and `second` were not static? (5)

4. For the purposes of this question, assume that Coordinates class has the following fields and constructor:

```
class Coordinates {
    public int x,y;
    public Coordinates(int x, int y){
        this.x = x;
        this.y = y;
    }
    ...
}
```

and that Circle class has the following fields:

```
class Circle {
    private double radius;
    private Coordinates coord;
    ...
}
```

- (a) Write a constructor `Circle(double r, Coordinates c)` for the Circle class. (5)

- (b) Override the `Object`'s method

```
public boolean equals(Object o)
```

for the `Coordinates` class so that it returns true if the current object has the same values of the instance fields as `o`, and false otherwise. (5)

- (c) Override the `Object`'s method

```
public boolean equals(Object o)
```

for the `Circle` class so that it returns true if the current object has the same values of the instance fields as `o`, and false otherwise. (5)

Question continued overleaf

- (d) Override the `Object`'s method

```
public Object clone()
```

for the `Circle` class so that it returns a 'deep copy' with all the instance fields having the same value (not the same reference) as in the current object. (5)

- (e) Explain the differences between variables of basic types and object (reference) types in Java with respect to assignment `=` and comparison `==`. (5)

5. Consider the following two classes:

```
class AClass {
    private String a;

    public AClass(String s) {
        a = new String(s);
    }
    protected String getA() {
        return a;
    }
    public void print() {
        System.out.println(a);
    }
}
class BClass extends AClass {
    private String b;

    public BClass(String s1, String s2) {
        super(s1);
        b = new String(s2);
    }
    public void print() {
        super.print();
        System.out.println(b);
    }
    public String getB() {
        return b;
    }
}
```

(a) Explain what is inheritance polymorphism. (5)

Question continued overleaf

- (b) Suppose that the following main() method has been added to the AClass class: (5)

```
public static void main(String[] args){
    BClass obj = new BClass("astring", "bstring");
    System.out.println(obj.getA());
}
```

Will this cause a compiler or run time error? If yes, explain why.

- (c) Will the following lines cause a compiler or run time error? If not, would both astring and bstring be printed, or only astring? Explain your answer. (5)

```
AClass obj1 = new BClass("astring", "bstring");
obj1.print();
}
```

- (d) Would the compiler report an error for the main() below? If yes, explain why. (5)

```
public static void main(String[] args){
    AClass obj2 = new BClass("astring", "bstring");
    obj2.getB();
}
```

- (e) Consider the following definitions: (5)

```
interface MyInterface {
    public Object getData();
}
class MyClass implements MyInterface {
    String name;
    public String getData() { return name;}
}
```

Would they compile? If not, why?

6. (a) The following piece of code throws an `ArrayIndexOutOfBoundsException`. Explain why and how this can be fixed. (5)

```
for (int i = 0; i < arr.length; i++){
    arr[i] = arr[i-1]*10;
}
```

- (b) The following piece of code throws a `NullPointerException`. Explain which line, why is the exception thrown and how this can be fixed. (5)

```
Object[] array = new Object[3];
for (int i = 0; i < array.length; i++){
    System.out.println(array[i].toString());
}
```

- (c) The `AbstractList` class is abstract, as the name suggests. It has a constructor `protected AbstractList()`. The following line causes a compiler error, explain what kind of error and why. (5)

```
AbstractList list = new AbstractList();
```

- (d) The following method definition causes several compiler errors: (10)

```
public static void readURL(String s) {
    URL url = new URL(s);
    BufferedReader in = new BufferedReader(
        new InputStreamReader(
            url.openStream()));

    String inputLine;
    while ((inputLine = in.readLine()) != null)
        System.out.println(inputLine);
    in.close();
}
```

Question continued overleaf

The errors are:

URLConnection.java:6: unreported exception
java.net.MalformedURLException; must be caught or declared
to be thrown

```
    URL url = new URL(s);  
                ^
```

URLConnection.java:9: unreported exception java.io.IOException;
must be caught or declared to be thrown

```
        url.openStream()));  
                ^
```

URLConnection.java:13: unreported exception java.io.IOException;
must be caught or declared to be thrown

```
        while ((inputLine = in.readLine()) != null)  
                ^
```

URLConnection.java:16: unreported exception java.io.IOException;
must be caught or declared to be thrown

```
        in.close();  
                ^
```

4 errors

Rewrite the method so that both exceptions are caught and handled. If a `MalformedURLException` is thrown, print to `System.err` "Malformed URL". If an `IOException` is thrown, print "IO Exception".