UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS
Department of Computer Science and Information Systems

End-of-Semester Exam

Academic Year:	2005/2006	**Semester:** Spring	
Module Title:	Programming Language 2 Data Structures & Algorithms	**Module Code:** CS5112 & CS4115	
Exam Duration:	2½ Hours	**Total Marks:** 70	
Lecturer:	Dr. N. S. Nikolov		

Instructions to Candidates:

Section A: Multiple-choice questions **15 marks**
ALL questions should be attempted.

Section B: Programming questions **55 marks**
ALL questions should be attempted.

Please write **ALL** answers in the answer booklet.
State clearly any assumptions you make.

Section A: Multiple-choice questions

All questions should be attempted

MULTIPLE CHOICE QUESTIONS ARE NOT AVAILABLE – SEE SECTION B ONLY

End of Section A

Section B: Programming questions

All questions should be attempted

Q6. (10 marks) Write a recursive method **`printK(int x, int k)`** which prints out those digits of the integer number **`x`** that are greater than or equal to **`k`**. For example,

`printK(29381,5)` prints out **`98`**
`printK(46392,4)` prints out **`469`**
`printK(10294,15)` prints out nothing

Q7. (30 marks) The area of a regular hexagon with side a is $\frac{3\sqrt{3}}{2}a^2$, and its perimeter is $6a$.

a. (5 marks) Write a class **`Hexagon`** which extends the abstract class **`Shape`**:

```
public abstract class Shape {
    public abstract double area( );
    public abstract double perimeter( );
}
```

b. (5 marks) Write class **`CompareHexagons`** which implements the interface **`Comparator<Hexagon>`** and provides a single method **`compare(Hexagon x1, Hexagon x2)`** which returns

- a positive number if the side of **`x1`** is larger than the side of **`x2`**.
- a negative number if the side of **`x1`** is smaller than the side of **`x2`**.
- zero, if **`x1`** and **`x2`** have sides with the same length.

c. (10 marks) Write a generic method

```
AnyType minelement(AnyType [] arr,
                   Comparator<? super AnyType> cmp)
```

which returns the smallest element in an array of elements of **`AnyType`** by using the **`compare(AnyType a, AnyType b)`** method of the comparator **`cmp`**.

d. (10 marks) Assume **`arr1`** is an array of hexagons sorted by the side of the hexagons in ascending order. Let **`arr2`** be another array of hexagons which is unsorted. Write a fragment of Java code that finds the smallest hexagon **`x`** in **`arr2`** by using the **`minelement`** method described above, and then checks whether there is a hexagon in **`arr1`** which is equal to **`x`**.

Q8. (15 marks) Consider the following class

```
class ListNode<AnyType>
{
   public ListNode(AnyType theElement) {
     this(theElement, null);
   }
   public ListNode(AnyType theElement, ListNode<AnyType> n) {
     element = theElement;
     next = n;
   }
   public AnyType element;
   public ListNode next;
}
```

a. (12 marks) Complete the implementation of the methods **`isEmpty()`, `push()`**, **`pop()`**, and **`top()`** in the class Stack.

```
public class ListStack<AnyType> implements Stack<AnyType>
{
    public boolean isEmpty(){ ... }
    public void push(AnyType x) { … }
    public void pop() { … }
    public void top() { … }

    private ListNode<AnyType> topOfStack = null;
}
```

b. (3 marks) Draw a diagram that shows the state of a stack at each step of the execution of the following sequence of operations:

`push(1), push(2), pop(), push(3), push(4), pop(), pop(), push(5)`.

End of Exam