DUBLIN CITY UNIVERSITY

SEMESTER ONE EXAMINATIONS 2008

MODULE: CA165
Computer Programming 1

COURSE: B.Sc. in Computer Applications
B.Sc. in Financial and Actuarial Mathematics
IFCCA International Foundation Certificate

YEAR: 1

EXAMINERS: Charlie Daly Ext: 5572

TIME ALLOWED: 3 Hours

INSTRUCTIONS:
Answer all questions in section A (40%).
Answer 2 question from section B (60%).
All questions in section A have equal marks.

Requirements for this paper

	Log Table
	Graph Paper
X	***Attached Answer Sheet***
	Statistical Tables
	Floppy Disk
	Actuarial Tables

THE USE OF PROGRAMMABLE OR TEXT STORING CALCULATORS IS EXPRESSLY FORBIDDEN

Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

(+ one appendix)

Section A (40%)

Answer all questions in section A; each question in section A is worth 5%.

Question 1

Expressions and Precedence

For each of the expressions below, write down the value of the expression. In addition, add parentheses to the expression so that it produces a different value and write down this new value.

Expression	Original value	New value
`10 * 2 + 3`		
`12.0 + 9 / 2`		
`10 + 8 % 3`		
`2 + 4 + "" + 2`		
`20 / 2 / 2`		

Question 2

Boolean expressions

Write down the value, ***true*** or ***false***, of the following boolean expressions.

Expression	Value
`5 == 5`	
`20 > 40 || 7 != 7`	
`5 <= 5 && 1 == 3/3`	
`!false && (false || true) && true`	
`(-1 < 0) == (18 < 0)`	

Question 3

The if statement

In the following two code fragments, assume that ***i*** and ***j*** are integers:

```
System.out.println("-");
if(i % j == 1)
   System.out.print("1");
if(j == 3)
   System.out.print("2");
```

Fragment A

```
System.out.println("-");
if(i % j == 1)
   System.out.print("1");
else if(j == 3)
   System.out.print("2");
```

Fragment B

Give values of `i` and `j` for which these two code fragments will print the *same* thing. What will be printed?

`i` ☐ `j` ☐ Output ☐

Give values of `i` and `j` for which these two code fragments will print *different* things. What will be printed?

`i` ☐ `j` ☐ Output of fragment A ☐

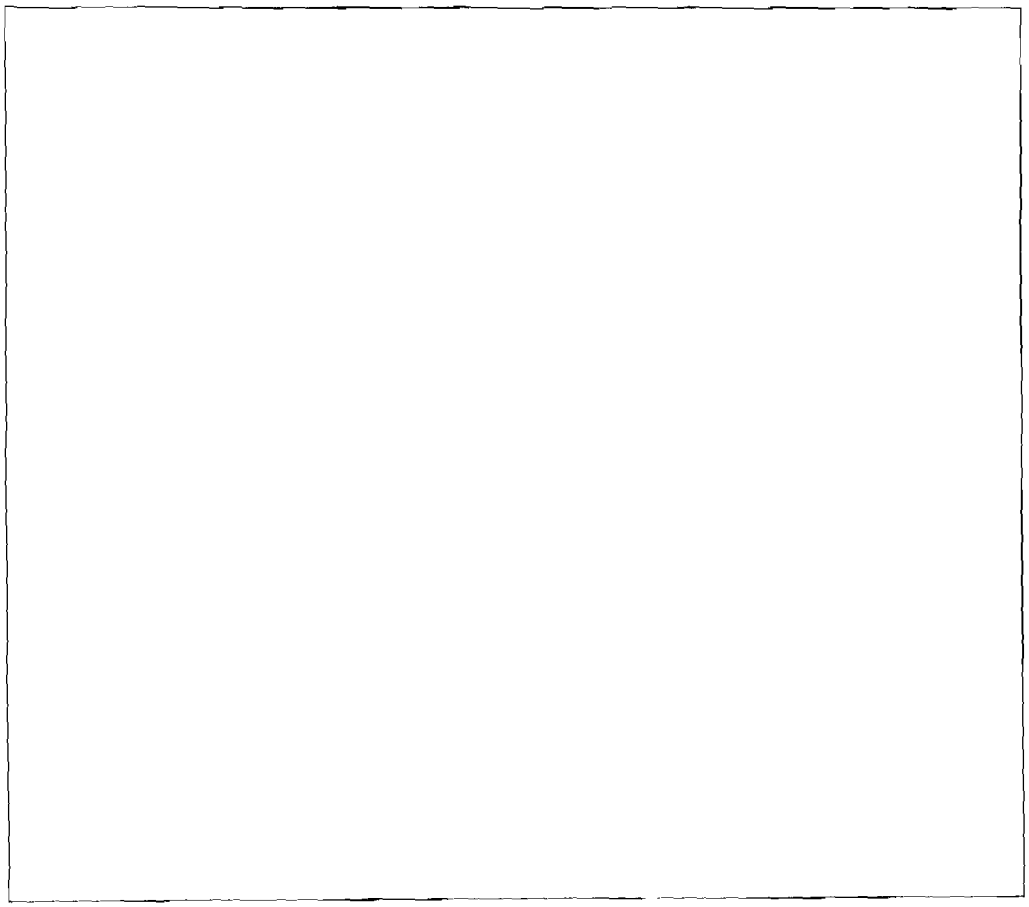
Output of fragment B ☐

Question 4

Loops

Use loops appropriately to print the following triangle of '*' symbols.

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



Question 5

Loops and characters

Consider the following code fragment:

```
String message = "Is this a door or is it the floor?";
String str = "";

int next;
next = 0;
next = next + 1;
while(next < message.length())
{
   if(message.charAt(next) == 'o' && next < message.length())
      next = next + 1;

   str = str + message.charAt(next);
   next = next + 1;
}
```

What value will ***str*** have when the code has executed?

Question 6

Tracing Code

Consider the following code:

```
int count = 0;

int [] a = { 8, 1, 10, 22, 4, 11, 22, 17, 38, 22, 8};

int max = 0;
for(int i = 1; i < a.length; i = i + 1)
   if(a[i] >= a[max])
   {
      count = count + 1;
      max = i;
   }
```

What will be the values of *max*, *a[max]* and *count* after the code has executed?

max

a[max]

count

Question 7

Sorting

The following code sorts the array ***numbers*** using the selection sort algorithm. You may assume that the swap method correctly swaps the two specified elements of the array ***numbers***.

```
for(i = 0; i < numbers.length - 1; i++)
{
   minIndex = i;
   for(j = i + 1; j < numbers.length; j++)
      if(numbers[j] < numbers[minIndex])
         minIndex = j;

   System.out.print(minIndex + ", ");
   swap(numbers, i, minIndex);
}
```

A print statement has been inserted just before the swap method call. What will be printed when the code is executed for each of the following values of ***num***:

num	Code output
{22, 3, 56, 4, 2, 4}	
{2, 4, 6, 8, 10, 12, 14}	
{10, 9, 8, 7, 6, 5, 4, 3}	

Question 8

Arrays, Strings and characters

```
String [] names = {"Kevin", "Cora", "Adam", "Sergio", "Atlas"};

for(int i = 0; i < names.length; i++)
{
   String s = names[i];
   int index = s.length() - 2;
   char c = s.charAt(index);
   System.out.print(c + " [" + s.length() + "] ");
}
```

What is the exact output produced by this code fragment?

Section B (60%)

Answer two questions; each question is worth 30%.

Question 1

a) Write a method called **shorten()** which takes a string and returns a string consisting of the first three characters. If the string has 3 or fewer characters, then the string should be returned unchanged.

[7 Marks]

b) A month code is a string of three lower case letters which represents a month. E.g. "jan" is the month code for "January". Write a method which will convert a string representing a month to its month code. Note that the method should ensure that all the month code letters are in lowercase. Note also that Strings have a method, **toLowerCase()**, which converts all uppercase letters to their lowercase equivalents.

[7 Marks]

c) Write a method which converts a month code to its corresponding number. A month code is the first three letters of the month all in lower case letters. E.g. the method should return 1 for "jan" and 12 for "dec". The method should take a string as a parameter which represents the month code and return and integer in the range 1-12 corresponding to month. If the string does not correspond to a valid month code, then the method should return -1. Note that you need to use the **equals** method to check if one string is equal to another, e.g. **s.equals("hello")**. will be true if **s** is "hello".

[8 Marks]

d) Write a method which takes a date and converts it to normalised numeric date. A normalised numeric date will be in the form dd/mm/yy. E.g. the first of February 2007 would be 01/02/07. The method will have three parameters to represent the date, the first will be an integer corresponding to the day of the month, the second will be a string representing the month and the third parameter will be an integer representing the year. The year will be an integer in the range 2000 to 2099.
The method invocation normaliseDate(1, "February", 2007) would return the string "01/02/07"

[8 Marks]

Question 2

Arrays

a) Write a method called `slideOver()` which will take an array parameter, `a`, and will move all the elements one location to the right. You should do nothing with the last element of the array and the first element of the array, `a[0]`, should be given a value of 0.
For example, the code

```
int [] a = {16, 12, 9, 30};
slideOver(a);
```

will cause the array to contain `{0, 16, 12, 9};`
All the elements have shifted one place to the right and the last element has gone.

[9 Marks]

b) Write a method called `moveOver()` which will take an array parameter, `a`, and an index, `start`, and will move the only elements from `start` to the end of the array minus 1. The first element moved, `a[start]`, should be given a value of 0 (naturally after it has moved up one place).
For example, the code

```
int [] a = {16, 12, 9, 30, 100, 200, 300, 400};
moveOver(a, 2);
```

will cause the array to contain `{16, 12, 0, 9, 30, 100, 200, 300};`

[6 Marks]

c) Write a method called `findLocation()` which takes as parameters a sorted array, `a`, and a value, `x`, will return the index of the leftmost element in the array which is larger than `x`. If no element is larger than x, return -1.
For example, the code

```
int [] a = {2, 6, 35, 67, 89, 100};
int location = findLocation(a, 40);
```

will cause `location` to have a value 3 as `a[3]`, i.e. 67 is the first element reading from the left which is larger than 40.

[9 Marks]

d) Write a method called `insert()` which will insert a value into a sorted array and shift the elements appropriately to make room for the new value. The rightmost element will be removed to make way for the new arrival.
For example, the code

```
int [] a = {2, 6, 35, 67, 89, 100};
insert(a, 40);
```

will cause the array to contain `{2, 6, 35, 40, 67, 89};`

[6 Marks]

Question 3

Consider the following infinite series

$$S = 1 + 1/2 + 1/4 + 1/8 + ...$$

That is, each successive term is half the previous term.

a) Write a method which takes an integer parameter, *n*, representing the number of terms to calculate and which calculates and returns the sum of these *n* terms.

[15 Marks]

b) Write a method which evaluates the above infinite series until the absolute value of the difference between two successive terms is less than $1 * 10^{-6}$. The method should return this value.

[15 Marks]

Appendix A

Documentation for selected methods of the String class

charAt

`public char charAt(int index)`

Returns the character at the specified index. An index ranges from `0` to `length() - 1`. The first character of the sequence is at index `0`, the next at index `1`, and so on, as for array indexing.

Parameters:

`index` - the index of the character.

Returns:

the character at the specified index of this string. The first character is at index `0`.

Throws:

IndexOutOfBoundsException - if the `index` argument is negative or not less than the length of this string.

endsWith

`public boolean endsWith(String suffix)`

Tests if this string ends with the specified suffix.

Parameters:

`suffix` - the suffix.

Returns:

`true` if the character sequence represented by the argument is a suffix of the character sequence represented by this object; `false` otherwise. Note that the result will be `true` if the argument is the empty string or is equal to this `String` object as determined by the `equals(Object)` method.

Throws:

NullPointerException - if `suffix` is `null`.

equals

`public boolean equals(Object anObject)`

Compares this string to the specified object. The result is `true` if and only if the argument is not `null` and is a `String` object that represents the same sequence of characters as this object.

Overrides:

equals in class Object

Parameters:

`anObject` - the object to compare this `String` against.

Returns:

`true` if the `String` are equal; `false` otherwise.

indexOf

`public int indexOf(int ch)`

Returns the index within this string of the first occurrence of the specified character. If a character with value `ch` occurs in the character sequence represented by this `String` object, then the index of the first such occurrence is returned -- that is, the smallest value *k* such that:

```
this.charAt(k) == ch
```

is `true`. If no such character occurs in this string, then `-1` is returned.

Parameters:
`ch` - a character.
Returns:
the index of the first occurrence of the character in the character sequence represented by this object, or `-1` if the character does not occur.

length

`public int length()`

Returns the length of this string. The length is equal to the number of 16-bit Unicode characters in the string.

Returns:
the length of the sequence of characters represented by this object.

substring

```
public String substring(int beginIndex,
                        int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`. Thus the length of the substring is `endIndex-beginIndex`.

Examples:

```
"hamburger".substring(4, 8) returns "urge"
"smiles".substring(1, 5) returns "mile"
```

Parameters:
`beginIndex` - the beginning index, inclusive.
`endIndex` - the ending index, exclusive.
Returns:
the specified substring.
Throws:
IndexOutOfBoundsException - if the `beginIndex` is negative, or `endIndex` is larger than the length of this `String` object, or `beginIndex` is larger than `endIndex`.

toLowerCase

`public String toLowerCase()`

Converts all of the characters in this String to lower case.

Returns:
the String, converted to lowercase.

CA165 Exam

Please fill in the details below and hand up your completed questions in Section A along with your script.

Seat Number								
Student Name								
Student ID								

Appendix 1