THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2006
Campus: City

COMPUTER SCIENCE

TEST

Principles of Programming

(Time allowed: 75 MINUTES)

NOTE: Attempt **ALL** questions
Write your answers in the space provided
There is space at the back for answers that overflow the allotted space
No calculators are permitted

Surname:	
Forenames:	
Student ID number:	
Login name:	

SURNAME: ..... FORENAMES: .....

CompSci 101 Test Results

Question	Marks	Out of
Question 1 (output)		20
Question 2 (methods)		7
Question 3 (methods)		7
Question 4 (arrays)		10
Question 5 (classes)		10
Question 6 (code tracing)		6
TOTAL		**60**

SURNAME: ..... FORENAMES: .....

Question 1 (20 marks)

a) What is printed by the following?

```
System.out.println("\\\\nnnn");
```

```
\\nnnn
```

(2 marks)

b) What is printed by the following?

```
System.out.println("1" + (2+3));
```

```
15
```

(2 marks)

c) What is printed by the following?

```
int value = 13;
boolean result1 = (value > 54) || !(value < 0);
boolean result2 = (value != 12 ) && (value%2 != 0);
System.out.println("result1 " + result1);
System.out.println("result2 " + result2);
```

```
result1 true
result2 true
```

(2 marks)

SURNAME: ..... FORENAMES: .....

d) What is printed by the following?

```
int[] numbers = {4, 2, -7, 5, 1, 6, 3};
int index = numbers[6] + numbers[numbers[4]];
System.out.println(numbers[numbers[index]]);
```

```
3
```

(2 marks)

e) What is printed by the following?

```
String word = "STARRY";
int position1 = word.indexOf('R');
int position2 = word.indexOf("AS");
System.out.println("position1 " + position1);
System.out.println("position2 " + position2);
```

```
position1 3
position2 -1
```

(2 marks)

f) What is printed by the following?

```
String name = "Computer Science";
String result = name.substring(3, 5) + name.substring(13);
System.out.println(result);
```

```
punce
```

(2 marks)

SURNAME: ..... FORENAMES: .....

g) What is printed by the following?

```
int value = 6;
if (value < 4) {
     System.out.println("Line 1");
     value = 7;
}
if (value > 6) {
     System.out.println("Line 2");
     value = -1;
} else if (value < 1) {
     System.out.println("Line 3");
} else {
     System.out.println("Line 4");
}
System.out.println("Line 5");
```

```
Line 4
Line 5
```

(2 marks)

h) What is printed by the following?

```
int value = 2;
if (value < 4) {
     System.out.println("Line 1");
     value = 7;
}
if (value > 6) {
     System.out.println("Line 2");
     value = -1;
} else if (value < 1) {
     System.out.println("Line 3");
} else {
     System.out.println("Line 4");
}
System.out.println("Line 5");
```

```
Line 1
Line 2
Line 5
```

(2 marks)

SURNAME: ..... FORENAMES: .....

i) What is printed by the following?

```
int x = Math.min(Math.min(5, 3), 4);
System.out.println(x);
```

```
3
```

(2 marks)

j) What is printed by the following?

```
int x = 3;
while (x > 0) {
     x--;
     System.out.println(x);
}
```

```
2
1
0
```

(2 marks)

SURNAME: ..... FORENAMES: .....

Question 2 (7 marks)

Write a method called `sameLastLetter()` which accepts two `String` parameters and returns a `boolean`. The method should return `true` if the last character of each of the two `String` parameters are the same (ignoring case). If the last characters are different, the method should return `false`. You can assume that the two `String` parameters both contain at least one character.

Note: in this method case should be ignored, e.g. the character 'a' should be considered the same as the character 'A'.

For example, executing the `Q2` application with the completed `sameLastLetter()` method should produce the following output:

```
> java Q2App
true
true
false
```

```
public class Q2 {

     public void start() {

          System.out.println(sameLastLetter("hiP", "hop"));
          System.out.println(sameLastLetter("hiPpy", "HAPPY"));
          System.out.println(sameLastLetter("bet", "bed"));
     }

     private  boolean  sameLastLetter( String word1,
                                       String word2 ){

          char last1, last2;
          word1 = word1.toUpperCase();
          word2 = word2.toUpperCase();

          int length1 = word1.length();
          int length2 = word2.length();
          last1 = word1.charAt(length1-1);
          last2 = word2.charAt(length2-1);

          if (last1 == last2) {
               return true;
          }
          return false;


     }
}
```

(7 marks)

CONTINUED

SURNAME: ..... FORENAMES: .....

Question 3 (7 marks)

Write a method `addToYear()` which accepts one `String` parameter and returns a `String`. The parameter is a `String` representing a date (consisting of a year and a month). The first 4 characters of this parameter String contain a 4-digit year. The method should return a `String` which is the same as the parameter `String` except that the year value is incremented by 1.

For example, executing the `Q3` application with the completed `addToYear()` method should produce the following output:

```
> java Q3App
2006_July
2002_April
2000_May
```

```
public class Q3 {

     public void start() {
          System.out.println(addToYear("2005_July"));
          System.out.println(addToYear("2001_April"));
          System.out.println(addToYear("1999_May"));
     }

     private  String  addToYear( String dateStr  ){


          String dateS;
          int date;
          dateS = dateStr.substring( 0, 4 );
          date = Integer.parseInt( dateS );
          date++;
          dateS = "" + date + dateStr.substring( 4 );

          return dateS;









     }
}
```

(7 marks)

CONTINUED

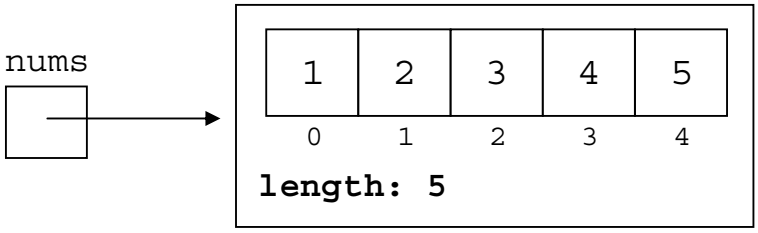
SURNAME: ..... FORENAMES: .....

Question 4 (10 marks)

a) Consider the array called `nums`, created as follows:

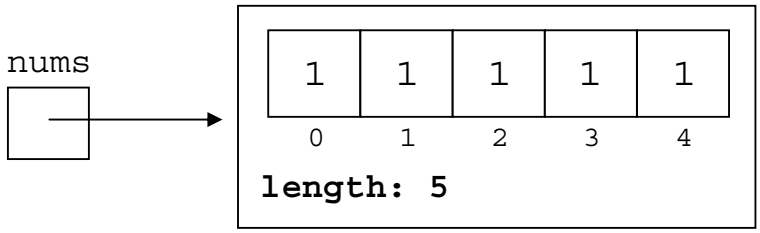
```
int[] nums = { 1, 2, 3, 4, 5 };
```

We can visualise this array as in the diagram below:



Complete the diagram of the array given below after the following loop has executed:

```
for (int i = 1; i < nums.length; i++) {
     nums[i] = nums[i-1];
}
```



(2 marks)

SURNAME: ..... FORENAMES: .....

b) Consider the following method in which some poor choices have been made for identifier names.

```
private int xxx(int[] nums) {
     int a = nums[0];
     int b = a;
     for (int i = 1; i < nums.length; i++) {
          if (nums[i] < a) {
               a = nums[i];
          } else if (nums[i] > b) {
               b = nums[i];
          }
     }
     return b - a;
}
```

What would be the output if this method is called as in the code segment below:

```
int[] nums = {-3, 4, 1, 7, 2, -4};
System.out.println(xxx(nums));
```

11

(2 marks)

Explain, in general, what value this method returns for any given array of integers.

The range of values in the array, in other words the difference between the largest and smallest values in the array

(1 mark)

Suggest a more appropriate identifier than `xxx` as the name of the method. The name you choose should indicate what the method does.

`calculateRange()` or `calculateLargestMinusSmallest()`

(1 mark)

SURNAME: ..... FORENAMES: .....

c) Write a method called `swapFirstAndLast()` which takes an array of integers as a parameter and swaps the values stored in the first and last positions of the array. For example, if this method has been defined correctly, the following program:

```
public class Test {

     public void start() {

          int[] a = {1, 2, 3, 4, 5};
          int[] b = {33, 44, 55};

          swapFirstAndLast(a);
          swapFirstAndLast(b);

          print(a);
          print(b);
     }

     private void swapFirstAndLast(int[] numbers) {
          // You need to define this method
     }

     private void print(int[] a) {
          for (int i = 0; i < a.length; i++) {
               System.out.print(a[i] + " ");
          }
          System.out.println();
     }
}
```

would produce the output:

```
5 2 3 4 1
55 44 33
```

Complete the `swapFirstAndLast()` method definition in the space provided below. You may assume that the array contains at least one element:

```
private void swapFirstAndLast(int[] numbers) {
```

```
     int temp = numbers[0];
     numbers[0] = numbers[numbers.length-1];
     numbers[numbers.length-1] = temp;
```

(4 marks)

```
}
```

CONTINUED

SURNAME: ..... FORENAMES: .....

Question 5 (10 marks)

a) Consider the `CandyBox` class, which represents a box of candy, defined as follows:

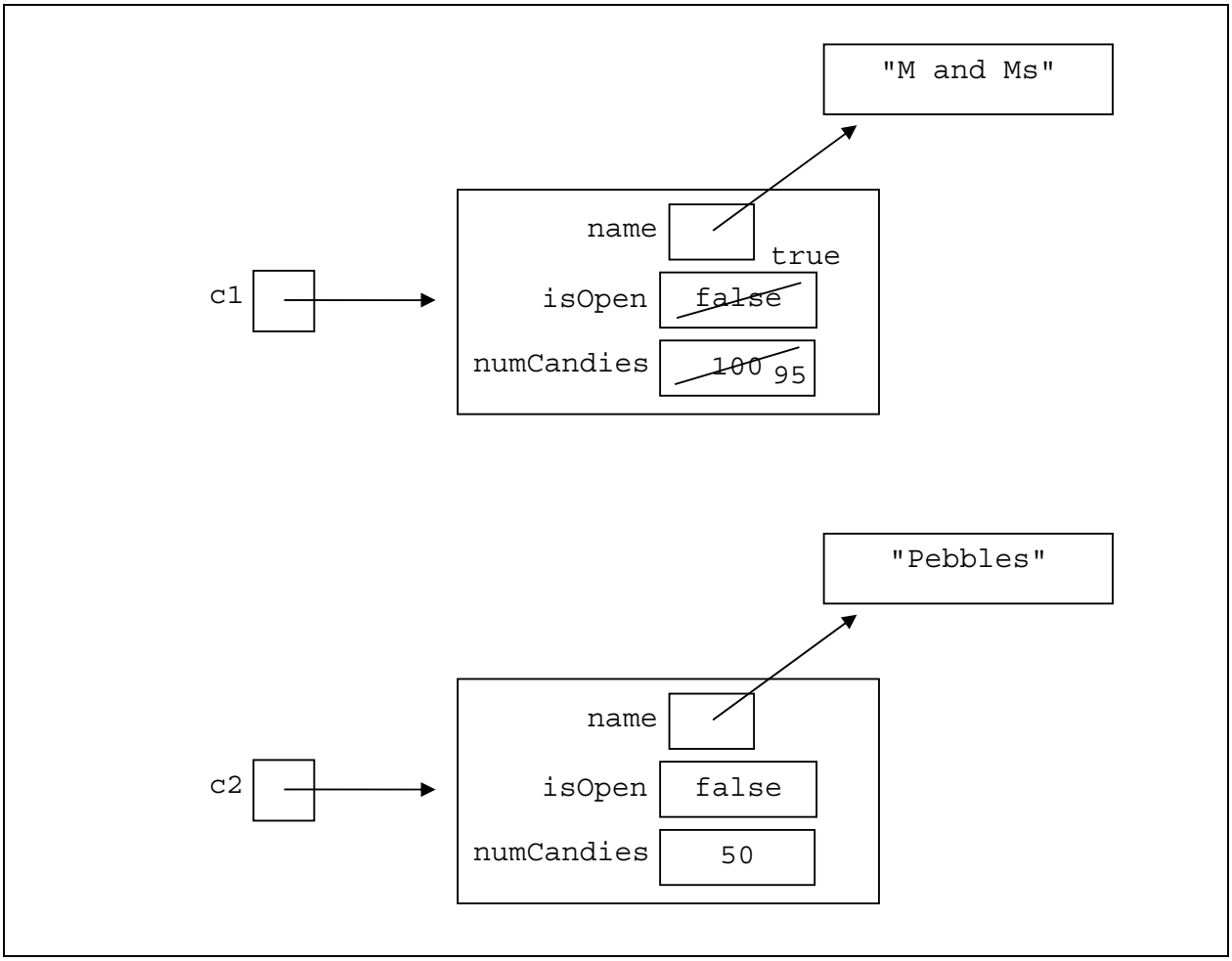
```
public class CandyBox {

     private String name;
     private boolean isOpen;
     private int numCandies;

     public CandyBox() {
          name = new String("M and Ms");
          numCandies = 100;
          isOpen = false;
     }

     public CandyBox(String n, int x) {
          name = n;
          numCandies = x;
          isOpen = false;
     }

     public void dispense() {
          if (isOpen) {
               numCandies = numCandies - 5;
          }
     }

     public void open() {
          isOpen = true;
     }

     public String toString() {
          return name + " " + isOpen + " " + numCandies;
     }
}
```

Two objects of type `CandyBox` are created as follows:

```
CandyBox c1 = new CandyBox();
CandyBox c2 = new CandyBox(new String("Pebbles"), 50);
```

Complete the diagram below illustrating the values that are stored in the instance variables for each of these objects. You should write very clearly on the diagram, as you are required to change the diagram when you complete part (b) of this question.

SURNAME: ..... FORENAMES: .....



(4 marks)

b) Given the two `CandyBox` objects created and initialised as in the diagram above, what would be the output after the following statements are executed? Note: you should also mark any changes to any instance variables clearly on the diagram above.

```
c1.open();

c1.dispense();
c2.dispense();

System.out.println( c1.toString() );
System.out.println( c2.toString() );
```

The output would be:

```
M and Ms true 95
Pebbles false 50
```

(6 marks)

CONTINUED

SURNAME: ..... FORENAMES: .....

Question 6 (6 marks)

What is the output when the following program is executed?

```
public class Q6 {

     public void start() {
          System.out.println();
          String value1 = method2(3, 4.2);
          int value2 = method1(25);
          boolean value3 = method3(0);
          System.out.println("value1: " + value1);
          System.out.println("value2: " + value2);
          System.out.println("value3: " + value3);
     }

     private int method1(int num1) {
          double num2 = num1 / 2.0;
          double num3 = 5.7;
          String word = method2(num2, num3);
          return Integer.parseInt(word);
     }

     private String method2(double num1, double num2) {
          double num3 = num1 + num2;
          String word = "" + num3;
          int position = word.indexOf(".");
          return word.substring(0, position);
     }

     private boolean method3(int value) {
          String word = method2(value, value);
          if(word.equals("0")) {
               return true;
          }
          return false;
     }
}
```

Show the output here:

```
> java Q6App



value1: 7
value2: 18
value3: true
```

(6 marks)

CONTINUED

SURNAME: ..... FORENAMES: .....

Working for Question 6:

Diagrams of the method calls are not required but partial credit may be given for working if the output is incorrect.

SURNAME: ..... FORENAMES: .....

OVERFLOW PAGE

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)

SURNAME: ..... FORENAMES: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)