THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2011
Campus: City

COMPUTER SCIENCE

TEST

Principles of Programming
(Time Allowed: 75 minutes)

Note:

- The use of calculators is NOT permitted.
- For Section A, use a dark pencil to mark your answers in the answer boxes on the Teleform sheet. There is one correct answer per question. Each question is worth 2 marks.
- For Section B, write your answers in the spaces provided in this booklet. Write as clearly as possible. The space provided will generally be sufficient but is not necessarily an indication of the expected length. Extra space is provided at the end of this exam book.

Surname:	
First Name(s):	
Student ID:	
Login Name (UPI):	
LAB TIME: *where your marked test will be returned*	*Day of week* *Starting time of lab*

— MARKERS ONLY —

Question	Marks	Out Of
1 - 20		40
21		23
22		22
23		15
Total		**100**

1) What is the output of the following code segment?

```
int a = 10;
int b = 20;
int c;

c = a;
a = b;
b = c;

System.out.println(a);
System.out.println(b);
```

(a)
```
20
20
```

(b)
```
10
20
```

(c)
```
20
10
```

(d)
```
10
10
```

(e) This code will not compile because `c` is not initialized to a literal value.

2) What output is produced when the following code is executed?

```
System.out.println(1 + 4 + 3 * 5 / 2 + " + " + 4 + 4);
```

(a) `12 + 44`

(b) `12.5 + 8`

(c) `11 + 44`

(d) `12.5 + 44`

(e) `12 + 8`

3) Consider the following set of conditional statements:

```
if (x % 2 == 1) {

      if ((x / 2 < 15) && (x > 10)) {
            System.out.println("TRUE");
      }

}
```

This segment of code will produce the output "TRUE" when x has what value?

(a) 31

(b) 5

(c) 12

(d) 21

(e) There is no value for x that will cause "TRUE" to be printed

4) What output is produced when the following code is executed?

```
int a = 16;
int b = 7;

int c = a % a;
int d = b % (b + 1);
int e = a % b;
int sum = c + d + e;

System.out.println("sum: " + sum);
```

(a) sum: 19

(b) sum: 9

(c) sum: 2

(d) sum: 3

(e) None of the above

5) What output is produced by the following code?

```
int x = 10;
int y = 5;

if (x < 10) {
      if (y != 5) {
            System.out.println("aaa");
      } else {
            System.out.println("bbb");
      }
} else if (y < 10) {
      System.out.println("ccc");
} else {
      if (y != 15) {
            System.out.println("ddd");
      } else {
            System.out.println("eee");
      }
}
```

(a) `bbb`

(b) `ccc`

(c) `eee`

(d) `ddd`

(e) `aaa`

6) What is the output of the following code?

```
String a, b, c;

a = new String("cat");
b = a;
a = new String("dog");
c = b;

System.out.println(a);
System.out.println(b);
System.out.println(c);
```

(a)
```
dog
dog
dog
```

(b)
```
cat
dog
dog
```

(c)
```
dog
cat
cat
```

(d)
```
dog
dog
cat
```

(e)
```
cat
cat
cat
```

7) Consider the following code (notice that the values assigned to variables `a` and `b` have been replaced with **???**):

```
boolean a = ???;
boolean b = ???;

boolean result = (!a || b) || b;
```

What values should be assigned to variables `a` and `b` such that the variable `result` will store the value **false**?

(a)
```
boolean a = true;
boolean b = false;
```

(b)
```
boolean a = false;
boolean b = false;
```

(c)
```
boolean a = true;
boolean b = true;
```

(d)
```
boolean a = false;
boolean b = true;
```

(e) None of the above

8) What output would be produced by the following code?

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

int a = nums[nums[1] + nums[3]];

System.out.println(a);
```

(a) `2`

(b) Nothing, because an out of bounds error (ArrayIndexOutOfBoundsException) would occur.

(c) `3`

(d) `4`

(e) `5`

9) What output is produced when the following code is executed?

```
int a = 4;
int b = a;
b++;

double d = a + b + 2;
double e = b / 2.0;

int c = (int) (d / 2 * 3);
d = (a + 1) / 2;

System.out.println("c: " + c  + " d: " + d  + " e: " + e);
```

(a) `c: 16 d: 2.5 e: 2.0`

(b) `c: 16 d: 2.0 e: 2.0`

(c) `c: 16 d: 2.5 e: 2.5`

(d) `c: 16 d: 2.0 e: 2.5`

(e) None of the above

10) Which of the following **could be** the output of the following section of code?

```
String word = "ABCDEFGHIJK";

int pos = (int)(Math.random() * (word.length() - 2));

word = word.substring(1, pos + 2);

System.out.println(word);
```

(a) `ABCDEFGHI`

(b) `ABCDEFGHIJ`

(c) `BCDEFGHIJK`

(d) `BCDEFGHIJ`

(e) None of the above

11) What output is produced when the following code is executed?

```
public void start() {
      String result = select("2. A E I O U");
      System.out.println(result);
}

private String select(String words1) {
      final String LETTERS = "AEIOU 0123456789";
      int pos;
      char c;
      String words2 = "";

      for (int i = 0; i < words1.length(); i++) {
            c = words1.charAt(i);
            pos = LETTERS.indexOf(c);

            if (pos == -1) {
                  words2 = words2 + c;
            } else if (pos > 5) {
                  words2 = words2 + 'o';
            } else if (pos < 5) {
                  words2 = words2 + '*';
            }
      }

      return words2;
}
```

(a) `*.ooooo`

(b) `oo*o*o*o*o*`

(c) `o.*****`

(d) `o.o*o*o*o*o*`

(e) None of the above

12) What output is produced when the following code is executed?

```
public void start() {
  method05(34, 41, 62, 11);
}

private void method05(int num1, int num2, int num3, int num4) {


  int num5, num6, num7;

  num5 = Math.max( Math.min(num1, num2), Math.min(num3, num4));

  num6 = Math.min( Math.max(num1, num2), Math.max(num3, num4));
  num7 = Math.min(num5, num6);

  System.out.println(num7);
}
```

(a) `34`

(b) `62`

(c) `11`

(d) `41`

(e) None of the above

13) What output is produced by the following code segment?

```
int sum = 0;
for (int i = 1; i <= 3; i++) {
     sum += i;
}
System.out.println(sum);
```

(a) `3`

(b) `6`

(c) `2`

(d) `0`

(e) `1`

14) Consider the following while loop. The value in the loop condition has been removed and replaced with **???**:

```
int i = 5;

while (i < ???) {
      System.out.println("HELLO");
      i = i + 2;
}
```

What value could replace the **???** so that the word "HELLO" is printed **exactly one time**.

(a) `10`

(b) `8`

(c) `9`

(d) `7`

(e) `5`

15) Consider the following segment of code:

```
int x = 5;
int y = 6;
boolean guess = true;

boolean a = x || guess;          // statement 1
boolean b = !guess && (x < y);   // statement 2
```

If a statement contains a *syntax error*, then it will not compile. Assuming we attempt to compile these statements, which of the following statements is **true**:

(a) statement 1 will not compile, but statement 2 would compile correctly

(b) both statements will compile, but there will be an error if the code is executed

(c) statement 1 would compile, but statement 2 will not compile correctly

(d) both statements will compile correctly, and the code will execute without error

(e) neither statement will compile correctly

16) The following program does not compile. What is the number of the line of code which causes the compile error.

```
1     private void aMethod(int number) {
2           final String SPACES = "                              ";
3           char c;
4           String digits = number;
5           int length = digits.length();
6
7           String filler1 = "";
8           String filler2 = SPACES.substring(0, length);

9           for(int i = length - 1; i >= 0; i--) {
10             c = digits.charAt(i);

11             System.out.println(filler2 + c + filler1);

12             filler1 = filler1 + 0;
13             filler2 = filler2.substring(0, filler2.length() - 1);
14          }
15    }
```

(a) 8

(b) 4

(c) 13

(d) 12

(e) 10

17) Which of the following code fragments is the **correct** way of declaring and creating a String array consisting of 21 elements?

(a)
```
String[] words;
words = new String[21];
```

(b)
```
String[] words = new String(21);
```

(c)
```
String words = new String[21];
```

(d)
```
String[] words;
words = new (21)String;
```

(e)
```
String[21] words = new String[];
```

18) Suppose the array `nums` is declared and initialized as follows:

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

What output is produced by the following code segment?

```
int sum = 0;

for (int i = 0; i <= nums.length; i++) {
     if (i % 2 == 1) {
          sum += nums[i];
     }
}

System.out.println(sum);
```

(a) `9`

(b) `0`

(c) `15`

(d) `6`

(e) Nothing, because an out of bounds error (ArrayIndexOutOfBoundsException) would occur.

19) What is the output of the following code?

```
public void start() {

      int a, x;

      a = 0;
      x = 0;

      changes(a, x);

      System.out.println("a: " + a);
      System.out.println("x: " + x);
}

private void changes(int a, int b) {
      a = 100;
      b = 200;
}
```

(a)
```
a: 0
x: 0
```

(b)
```
a: 100
x: 0
```

(c)
```
a: 100
x: 200
```

(d)
```
a: 0
x: 200
```

(e) this will not compile as the names of the formal parameters do not match the names of the actual parameters

20) Consider the following definition of a `Car` class:

```
public class Car {

      private String make;
      private int speed;

      public Car(String newMake) {
            make = newMake;
            speed = 0;
      }

      public void accelerate(int howMuch) {
            speed += howMuch;
      }

      public int getSpeed() {
            return speed;
      }
}
```

What would be produced as output when the following start() method is executed?

```
public void start() {

      Car a, b;

      a = new Car("A");
      b = a;
      a.accelerate(10);
      b.accelerate(25);

      System.out.println(a.getSpeed());
      System.out.println(b.getSpeed());
}
```

(a)
```
35
35
```

(b)
```
10
25
```

(c)
```
0
0
```

(d)
```
25
10
```

(e)
```
25
25
```

SECTION B

Answer all questions in this section in the space provided. If you run out of space then please use the Overflow Sheet and indicate in the allotted space that you have used the Overflow Sheet.

Question 21 [23 marks]

a) Complete the method header for the following method (i.e. complete the first line of the method definition). The method, `methodA()`, is called in the following way:

```
int result = methodA(24, "minus", "4");
```

```
private ___int__ methodA( ___int first, String operator,

                                    ___ String second ____ ) {
     if (operator.equals("plus")) {
          return first + Integer.parseInt(second);
     }
     return first - Integer.parseInt(second);
}
```

(3 marks)

b) Complete the method header for the following method (i.e. complete the first line of the method definition). The method, `methodB()`, is called in the following way:

```
boolean result = methodB("*");
```

```
private ___boolean____ methodB( ___String symbol________

                              ___________________________________ ) {
     final String OPERATORS = "+-*/";
     int pos = OPERATORS.indexOf(symbol);
     return pos > -1;
}
```

(3 marks)

c) Complete the method header for the following method (i.e. complete the first line of the method definition). The method, `methodC()`, is called in the following way:

```
methodC("Mister", true, 'B');
```

```
private __void___ methodC( _String title, boolean inShortForm,

                           _______char initial_____________ ) {
     int pos = title.length() - 1;
     if (inShortForm) {
          System.out.println("" + title.charAt(0) +
                     title.charAt(pos) + ". " + initial + " ");
     } else {
          System.out.println(title + initial + " ");
     }
}
```

(3 marks)

d) Complete the `getDifference()` method which returns the *absolute difference* between the two numbers passed as parameters. Note: the absolute difference between two numbers is a positive value.

```
private int getDifference(int first, int second) {









     return Math.abs(first - second);




}
```

(4 marks)

e) Complete the `getARandomLetter()` method which returns a single character. The character returned by the method is randomly selected from the letters of the String passed to the method as a parameter. Note: each letter of the parameter String should have an equal chance of being returned by the method.

```
private char getARandomLetter(String letters) {









     int r = (int)(Math.random() * letters.length());
     return letters.charAt(r);

}
```

(4 marks)

f) The `ConvertMinutes` program prompts the user for the number of minutes, reads the number entered by the user, converts the minutes into days, hours and minutes, and finally prints the information. For example, if the user enters 76 minutes, the program displays "Days: 0, hours: 1, minutes: 16".

Below are THREE examples showing possible program output (user input is in bold):

```
Enter minutes: 76
Days: 0, hours: 1, minutes: 16

Enter minutes: 3000
Days: 2, hours: 2, minutes: 0

Enter minutes: 24
Days: 0, hours: 0, minutes: 24
```

Write your code in the `start()` method below.

```
public class ConvertMinutes {

     public void start() {
          int minutes, hours, days;









          System.out.print("Enter minutes: ");

          minutes = Integer.parseInt(Keyboard.readInput());

          hours = minutes / 60;
          days = hours / 24;
          minutes = minutes % 60;
          hours = hours % 24;









          System.out.println("Days: " + days + ", hours: "
                             + hours + ", minutes: " + minutes);
     }
}
```

(6 marks)

Question 22 **[22 marks]**

a) Complete the `smallestPosition()` method below. This method should return the *index position* of the smallest value stored in an array of integers.

For example, given the array:

```
int[] nums = { 6, 4, 3, 8, 7 };
```

the method call:

```
smallestPosition(nums);
```

would return the value:

```
2
```

because the smallest value in the array is in index position 2.

You can assume that the input array, which is passed to the method as a parameter, will contain at least one value.

```
private int smallestPosition(int[] nums) {









    int smallestIndex = 0;

    for (int i = 1; i < nums.length; i++) {
         if (nums[i] < nums[smallestIndex])
              smallestIndex = i;
    }
    return smallestIndex;









}
```

(6 marks)

b) Complete the `leftmostPosition()` method below. This method is passed two parameters: an array of integers and an integer value. The method should return the *index position* of the first occurrence (from left to right) of the given value inside the array. If the given value is not contained in the array, the method should return -1.

For example, given the array:

```
int[] nums = { 4, 3, 2, 3, 7 };
```

the method calls:

```
leftmostPosition(nums, 3)
leftmostPosition(nums, 8)
```

would return the values:

```
1
-1
```

because the first occurrence of the value 3 in the array is in index position 1, and the value 8 does not appear in the array.

You can assume that the array will contain at least one element.

```
private int leftmostPosition(int[] nums, int value) {









    for (int i = 0; i < nums.length; i++) {
         if (nums[i] == value) {
              return i;
         }
    }
    return -1;









}
```

(7 marks)

c) *Complete* the `combineArrays()` method below. This method is passed two arrays of integers as parameters. You can assume that each array contains at least one value.

The method should return a *new array* that contains all of the elements of the first array *followed by* all of the elements of the second array (the variable for this new array, `nums3`, has been declared in the method for you).

Note: the length of `nums3`, the array which is returned by this method, should be the sum of the lengths of the two input arrays.

```
private int[] combineArrays(int[] nums1, int[] nums2) {

     int[] nums3;



     nums3 = new int[nums1.length+nums2.length];

     for (int i = 0; i<nums1.length; i++) {
          nums3[i] = nums1[i];
     }

     for (int i = 0; i<nums2.length; i++) {
          nums3[nums1.length + i] = nums2[i];
     }

     return nums3;









}
```

(9 marks)

Question 23 [15 marks]

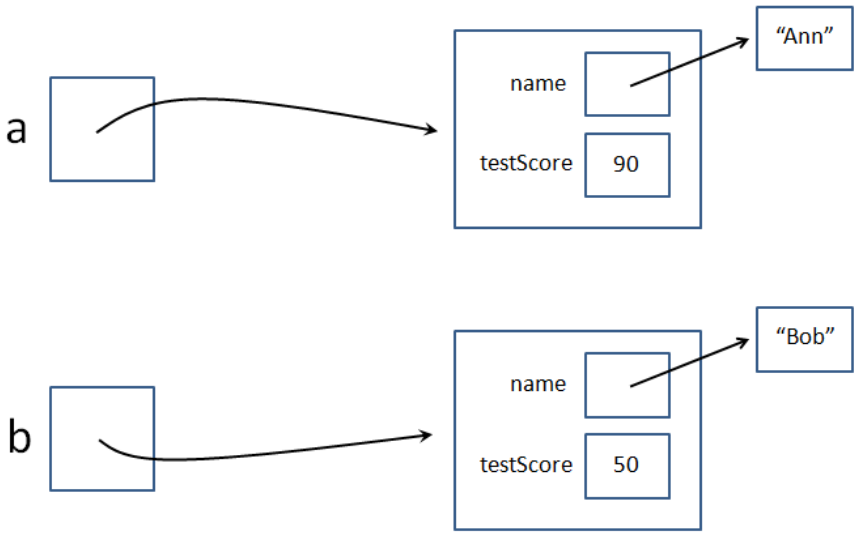
Consider the following program which creates two `Student` objects, sets the test score for each object, and then displays the scores.

```
public void start() {

     Student a, b;

     a = new Student("Ann");
     b = new Student("Bob");

     a.setTestScore(90);
     b.setTestScore(50);

     System.out.println("Ann: " + a.getTestScore());
     System.out.println("Bob: " + b.getTestScore());
}
```

The output of this program would be:

```
Ann: 90
Bob: 50
```

You could visualise these two objects as follows:



Complete the definition of the `Student` class in the spaces provided below:

```
public class Student {

     private String name;
     private int testScore;

     // Define the constructor method below
```

```
public Student(String initName) {
     name = initName;
     testScore = 0;
}
```

(5 marks)

```
     // Define the setTestScore method below
```

```
public void setTestScore(int score) {
     testScore = score;
}
```

(5 marks)

```
     // Define the getTestScore method below
```

```
public int getTestScore() {
     return testScore;
}
```

(5 marks)

```
}
```

Overflow page – please number answers carefully