THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2006
Campus: City

T E S T

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: ONE hour and FIFTEEN minutes)

NOTE: **Attempt ALL questions.**
Answer the multiple choice questions in section A by circling the correct answer on the answer booklet.
Write the answers to the questions in section B in the space provided on the answer booklet.
Calculators are NOT permitted.

SECTION A: MULTIPLE CHOICE QUESTIONS

Each question in this section is worth 3 marks. Circle the letter corresponding to your choice on the attached answer sheet. There is only one correct answer for each question.

1. What is the output produced by the following code?

```
double d = 2 * (5 / 2) + 5 / 2.0;
System.out.println(d);
```

(a) `4.5`

(b) `6.0`

(c) `7.0`

(d) `7.5`

(e) none of the above

2. What is the output produced by the following code?

```
int a = 1;
int b = 2;
int c = 3;

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

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

(a) `2 2 1`

(b) `2 2 3`

(c) `2 3 1`

(d) `2 3 2`

(e) `2 3 3`

3. What is the output produced by the following code?

```
String word1 = new String("PERFECTION");
String word2 = new String("ON");
String word3 = word1.toLowerCase();

System.out.println("1 " + word1);

if (word3  ==  word1) {
     System.out.println("2 " + word1);
}
```

(a) `1 perfection`

(b) `1 perfection`
`2 perfection`

(c) `1 PERFECTION`
`2 PERFECTION`

(d) `1 PERFECTION`

(e) none of the above

4. Which of the following symbolic constant identifiers correctly adheres to the naming conventions covered in this course?

(a) `final int numberOfStudents = 480;`

(b) `final int NUMBEROFSTUDENTS = 480;`

(c) `final int NumberOfStudents = 480;`

(d) `final int number_Of_Students = 480;`

(e) `final int NUMBER_OF_STUDENTS = 480;`

5. What is the output produced by the following code?

```
String number = "0";
System.out.println(1 + 1 + number + 1 + 1);
```

(a) `2011`

(b) `11011`

(c) `202`

(d) `1102`

(e) none of the above

6. Consider the following statement:

```
int rand = (int)(Math.random() * 20) + 10;
```

What is the correct range of values (**including** the numbers shown) that may be stored in the variable `rand` after this statement has executed?

(a) `10 to 29`

(b) `10 to 30`

(c) `11 to 29`

(d) `11 to 30`

(e) `10 to 20`

7. In CompSci 101 we always make our instance variables:

(a) `public`

(b) `protected`

(c) `public final static`

(d) `private`

(e) `private final static`

8. What is the output produced by the following code?

```
int sum = 0;
int number = 50;

if (number > 40) {
     sum++;
     if (number <= 50) {
          sum++;
     }
     sum++;
}

if (number%2 == 1) {
     sum++;
}
sum++;

if (number > 50) {
     sum++;
}
System.out.println("sum: " + sum);
```

(a) `sum: 3`

(b) `sum: 4`

(c) `sum: 5`

(d) `sum: 6`

(e) none of the above

9. What is the output produced by the following code?

```
int answer = 0;
for (int i = 1; i <= 10; i++) {
     for (int j = i; j > 0; j--) {
          answer++;
     }
}
System.out.println(answer);
```

(a) `55`

(b) `45`

(c) `100`

(d) `99`

(e) `10`

10. What is the output produced by the following code?

```
String words = "HE LED THE WAY";
int position1 = words.indexOf("HE");
int position2 = words.indexOf('L');
char c = words.charAt(position1);
int sum = position1 + position2;
System.out.println("sum: " + sum);
System.out.println("c: " + c);
```

(a)
```
sum: 3
c: H
```

(b)
```
sum: 2
c: H
```

(c)
```
sum: 3
c: E
```

(d)
```
sum: 4
c: E
```

(e) none of the above

11. An array initialiser statement, such as:

```
boolean[] bits = { true, false, true, false, true, true };
```

does *three* things in the following order, it:

(a) modifies, assigns, and constructs

(b) declares, constructs, and initialises

(c) declares, initialises, and modifies

(d) constructs, declares, and instances

(e) constructs, initialises, and constructs

12. What error message is displayed when the following statements execute:

```
int answer = 0;
int[] results = new int[21];
for (int i = 0; i <= 21; i++) {
     answer = answer / results[i] + 1;
}
```

(a) `VariableNotInitialisedException`

(b) `ArrayIndexOutOfBoundsException`

(c) `NullPointerException`

(d) `ArithmeticException`

(e) `ArrayStoreException`

13. What is stored in the array b after the following statements are executed:

```
String[] a = {"one", "two", "three", "four"};
String[] b = a;
b[1] = a[3];
a[2] = a[1];
b[0] = "five";
```

(a) "one", "two", "three", "four"

(b) "five", "four", "three", "four"

(c) "five", "three", null, null

(d) "five", "four", "two", "four"

(e) "five", "four", "four", "four"

14. What is the output produced by the following code?

```
String phrase = "BITS AND PIECES";
String part1 = phrase.substring(9);
String result = phrase.substring(0, 3) + " " + part1;
System.out.println(result);
```

(a) BI PIECE

(b) BIT P

(c) IT PIECES

(d) BIT PIECES

(e) BITS PIECES

15. Given the following definition of the method called `methodD()`:

```
private String methodD(int number1, double number2,
                                              boolean isActive){
     if (number2 > number1 && isActive) {
          return "true";
     }
     return "false";
}
```

Which of the following method call statements compiles without error?

(a) `String result = methodD(30, 60, true);`

(b) `String result = methodD(60, 30, "true");`

(c) `String result = methodD(30.5, 15, false);`

(d) `String result = methodD("12", 60.6, false);`

(e) `boolean result5 = methodD(30, 16.6, true);`

SECTION B

Question 16 (7 marks)

The simple program below is supposed to produce the output:

```
10+20=30
```

However, there are 3 syntax errors in the code which prevent it from compiling. There is also one logic error which would cause the program to produce unexpected output.

Locate and circle these 3 syntax errors and the logic error. You do not need to provide corrections for the 3 syntax errors, but you do need to provide a correction for the logic error.

```
public void class Addition {

     public void start() {
          int a = 10;
          int b = 20;
          int c = sum(a, b)
          System.out.println(a + b + "=" + c);
     }

     private sum(int x, int y) {
          return x + y;
     }
}
```

(7 marks)

Question 17 (6 marks)

What is the output produced when the `start()` method below is executed?

```
public void start() {

     boolean result1, result2;
     int value = 5;
     String word = "HUNGRY";
     result1 = false;

     result2 = word.length()>5 && (value%2==3 || result1);
     System.out.println("1. " + result2);

     result2 = (word.length()>5 || value%2==0) && ! result1;
     System.out.println("2. " + result2);

     result1 = !result1;
     System.out.println("3. " + result1);

}
```

(6 marks)

Question 18 (10 marks)

Consider the following program (with some code missing) for calculating the number of boxes required to pack a given number of cans.

There are two different sized boxes that can be used for packing the cans: a big box which can hold 50 cans and a small box which can hold 12 cans (symbolic constants have been declared to store these values).

The user inputs the number of cans in total that need to be packed. The program should calculate the minimum number of boxes required, so the big boxes should always be used before the small boxes. There also cannot be any partially filled boxes – the boxes must be completely full of cans. The number of left over cans which cannot be packed in any box should also be displayed.

You need to complete the code that is missing from the empty boxes. **You must not use any literal numeric values in your code** – refer to variable and symbolic constant identifiers only.

```
public class CalculateBoxesRequired {
     public void start() {

          final int CANS_PER_BIG_BOX = 50;
          final int CANS_PER_SMALL_BOX = 12;

          String input;

          int totalCans;
          int numBigBoxes;
          int numSmallBoxes;
          int leftOverCans;

          System.out.print("How many cans do you need to pack? ");

          input = Keyboard.readInput();

          totalCans =
          numBigBoxes =
          leftOverCans =
          numSmallBoxes =
          leftOverCans =

          System.out.println("NUMBER OF BOXES:");
          System.out.println("Big: " + numBigBoxes);
          System.out.println("Small: " + numSmallBoxes);
          System.out.println("Left over cans: " + leftOverCans);
     }
}
```

An example of the program running is given below (the user input is shown in bold):

```
How many cans do you need to pack? 90
NUMBER OF BOXES:
Big: 1
Small: 3
Left over cans: 4
```

(10 marks)

Question 19 (6 marks)

What is the output produced when the `start()` method below is executed?

```
public void start() {
     lotsOfIfs(16, "OUT");
}

private void lotsOfIfs(int value, String word) {
     if (!word.equals("SING")) {
          System.out.println("Line 1");
          value = 12;
          word = word + " LOUD";
     } else if (value > 10) {
          System.out.println("Line 2");
          value = value/2;
          word = word + word + word;
     }
     if (value <= 6) {
          System.out.println("Line 3");
          value = 5;
     } else if (word.length() > 4) {
          System.out.println("Line 4");
     } else {
          return;
     }

     System.out.println("Line 6 " + word);
}
```

(6 marks)

Question 20 (10 marks)

Complete the method `getHoursAndMinutes()` which accepts an `int` parameter and returns a `String`. The parameter is the number of minutes. The method should convert the number of minutes to hours and minutes and return a `String` in the form: `hours:minutes`. For example, the method should return the `String` "`1:03`" if the number of minutes is 63.

Note: The resulting number of minutes always has two digits.

For example, executing the program below with the completed `getHoursAndMinutes()` method produces the following output:

```
> java TestApp
1:40
3:01
0:22
1:03
```

```
public class Test {

     public void start() {
          System.out.println(getHoursAndMinutes(100));
          System.out.println(getHoursAndMinutes(181));
          System.out.println(getHoursAndMinutes(22));
          System.out.println(getHoursAndMinutes(63));
     }

     private __________ getHoursAndMinutes(____________________) {

          // Complete this method
     }
}
```

(10 marks)

Question 21 (6 marks)

Complete the method below which is passed 2 parameters - an array of integers and a value to search for in the array. The method should return the index of the first element that contains the value specified in the parameter, or -1 if the value does not appear in the array.

NOTE: You can assume the array that is passed to the method as a parameter contains at least one element. Please also note that **you MUST use a `for` loop** to access the elements of the array.

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

     // Returns the first index at which value is stored in
     // the array, or -1 if it is not in the array

     // You need to write the code for this method
}
```

If you have completed the code for this method correctly, then the following statements:

```
int[] numbers = {47, 45, 81, 4, 16, 78, 18, 14};
int[] someMoreNumbers = {-1, -2, -3, -4, -5};

int index1 = findValue(numbers, 4);
int index2 = findValue(someMoreNumbers, -6);

System.out.println(index1);
System.out.println(index2);
```

should produce the output:

```
3
-1
```

(6 marks)

Question 22 (10 marks)

The `Computer` class keeps track of specifications for a computer. It records the CPU speed, the amount of RAM and the total storage capacity of its hard-drives.

If the `Computer` class has been defined correctly, the following code:

```
Computer computer1 = new Computer(2.0, 1024, 128);
Computer computer2 = new Computer(3.2, 2048, 256);

System.out.println(computer1);
System.out.println(computer2);

computer1.upGradeMemory(1024);
computer1.replaceCPU(3.2);

computer2.upGradeMemory(1024);
computer2.replaceCPU(3.4);

System.out.println(computer1);
System.out.println(computer2);
```

should produce this output:

```
2.0GHz 1024MB 128GB storage
3.2GHz 2048MB 256GB storage
3.2GHz 2048MB 128GB storage
3.4GHz 3072MB 256GB storage
```

You need to complete the following definition of the `Computer` class:

```
public class Computer {

    private double speed;
    private int memory;
    private int diskSpace;

    public Computer(double initSpeed, int initMemory,
                                              int initDiskSpace) {
        // complete
    }

    public String toString() {
        // complete
    }

    public void upGradeMemory(int extraMemory) {
        // complete
    }

    public void replaceCPU(double newSpeed) {
        // complete
    }
}
```

(10 marks)