

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2012
Campus: City

COMPUTER SCIENCE

TEST

Principles of Programming

(Time Allowed: 75 minutes)

- NOTE:
- No calculators are permitted
 - Compare the test version number on the Teleform sheet supplied with the version number above. If they do not match, ask the test supervisor for a new sheet.
 - Enter your name and student ID on the Teleform sheet. Your name should be entered left aligned. If your name is longer than the number of boxes provided, truncate it.
 - Answer Section A (Multiple choice questions) on the Teleform answer sheet provided. Answer Section B in the space provided in this booklet.
 - Use a dark pencil to shade in your answers in the multiple choice answer boxes on the Teleform sheet. Check that the question number on the sheet corresponds to the question number in this question book. If you spoil your sheet, ask the supervisor for a replacement.
 - There is space at the back for answers that overflow the allotted space.

Surname	
Forenames	
Student ID	
Login (UPI)	
Lab time	

Q1 – Q14 (/28)	Q17 (/10)	Q20 (/15)
Q15 (/12)	Q18 (/15)	Total /100
Q16 (/12)	Q19 (/8)	

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

SECTION A**MULTIPLE CHOICE QUESTIONS**

Each question in this section is worth 2 marks. For each question, choose the best answer according to the information presented in lectures. Select your preferred answer on the Teleform answer sheet by shading in the appropriate box.

Question 1

[2 marks] What is the output of the following code?

```
int a = 5;  
int b = a;  
int c = b - 3;
```

```
a = b + c * 2;  
b = b + 4;  
c = a / c;  
b++;
```

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

- (a) a 9 b 10 c 4.5
- (b) a 10 b 10 c 4
- (c) a 9 b 10 c 4**
- (d) a 10 b 10 c 4.5
- (e) none of the above

Question 2

[2 marks] What is the output of the following code?

```
int a = 5;
int b = a % 10;
int c = a % 5;
double d = a / 2;

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

- (a) b 5 c 0 d 2.5
- (b) b 5 c 5 d 2.0
- (c) b 5 c 0 d 2.0**
- (d) b 0 c 0 d 2.5
- (e) none of the above

Question 3

[2 marks] Which of the following could NOT be the output generated by the following code?

```
int number;
for (int i = 0; i < 5; i++) {
    number = (int) (Math.random() * 4 + 2);
    System.out.print(number);
}
```

- (a) 2 3 4 5 6**
- (b) 2 2 2 3 3
- (c) 3 4 4 3 3
- (d) 5 5 5 5 5
- (e) all of the above could be the output

Question 4

[2 marks] What is the output of the following code?

```
String word = "SWEET";  
char c1 = word.charAt(word.length() - 1);  
char c2 = word.charAt(1);  
System.out.println(c1 + ", " + c2);
```

- (a) T, W
- (b) E, W
- (c) T, S
- (d) E, S
- (e) none of the above

Question 5

[2 marks] What is the output of the following code?

```
String word = "SUPER";  
int length = word.length();  
int pos = length - 3;  
String part1 = word.substring(pos + 1);  
String part2 = word.substring(0, pos);  
System.out.println(part1 + part2);
```

- (a) SUER
- (b) SUPER
- (c) ERSU
- (d) PERSU
- (e) none of the above

Question 6

[2 marks] What is the output of the following code?

```
String word = "abracadabra";
int pos1 = word.indexOf('a');
int pos2 = word.indexOf('b');

if (pos1 > pos2) {
    System.out.println(pos1);
} else {
    System.out.println(pos2);
}
```

- (a) 2
- (b) -1
- (c) 1**
- (d) 0
- (e) none of the above

Question 7

[2 marks] What is the output when the following call:

```
printNumber(2, 4, 1, 3);
```

is made to the `printNumber()` method below?

```
private void printNumber(int n1, int n2, int n3, int n4){  
  
    int num1 = Math.min(n1, Math.min(n2, n3));  
    int num2 = Math.max(n1, Math.max(n2, n3));  
    int num3 = Math.max(num1, Math.max(num2, n4));  
  
    System.out.println(num3);  
}
```

- (a) 3
- (b) 2
- (c) 4**
- (d) 1
- (e) none of the above

Question 8

[2 marks] The following method takes two inputs: an array of integers and an integer.

```
private int mystery(int[] values, int x) {  
    for (int i = 0; i < values.length; i++) {  
        if (values[i] == x) {  
            return i;  
        }  
    }  
    return -1;  
}
```

Which of the following statements best describes the purpose of this method?

- (a) It returns all of the index positions where the value x is stored in the array, or -1 if the array has no elements
- (b) It returns the value x if x is stored in the array, or -1 if the array has no elements
- (c) It returns the value stored at index position x in the array, or -1 if x is not valid
- (d) It returns a count of the number of times the value x appears in the array, or -1 if x is not stored in the array
- (e) It returns the index position of the first occurrence of the value x in the array, or -1 if x is not stored in the array**

Question 9

[2 marks] What is the output of the following code?

```
public void start() {  
  
    int a, b;  
  
    a = 2;  
    b = methodA(a);  
  
    System.out.println(a + " " + b);  
}  
  
private int methodA(int a) {  
    int b = 3;  
    a = a + 5;  
    return a + b;  
}
```

(a) 7 10

(b) 2 10

(c) 2 2

(d) 2 3

(e) 7 3

Question 10

[2 marks] What is the output of the following code?

```
boolean b = (4 < 4);  
boolean c = (3 != 4);  
boolean d = (3 > 4);  
  
System.out.println( (b || c) && (c || d) );  
System.out.println( (b && c) || (c && d) );
```

- (a) true
false
- (b) false
false
- (c) true
true
- (d) false
true
- (e) It is not valid to use the && operator in this way

Question 11

[2 marks] What is the output of the following code?

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

if (a > 10) {
    System.out.println("one");
} else if (b == 20) {
    System.out.println("two");
} else if (a <= 10) {
    System.out.println("three");
} else {
    System.out.println("four");
}
```

- (a) two
- (b) one
- (c) three
- (d) four
- (e) no output

Question 12

[2 marks] What is the output of the following code?

```
String word1 = new String("Hello");
String word2 = new String("Hello");
String word3 = word1;

System.out.println("1: " + (word1 == word2));
System.out.println("2: " + (word1 == word3));
System.out.println("3: " + (word1.equals(word2)));
System.out.println("4: " + (word1.equals(word3)));
```

(a) 1: false
2: true
3: true
4: true

(b) 1: false
2: false
3: true
4: true

(c) 1: true
2: true
3: false
4: false

(d) 1: true
2: false
3: false
4: false

(e) 1: true
2: true
3: true
4: false

Question 13

[2 marks] What is the output of the following code?

```
public void start() {
    int number = 100;
    addSome(number);
    System.out.println("start: " + number);
}

private void addSome(int number) {
    number = number + 122;
    System.out.println("inTheAddSomeMethod: " + number);
}
```

(a) inTheAddSomeMethod: 222
start: 222

(b) inTheAddSomeMethod: 222
start: 100

(c) start: 222
inTheAddSomeMethod: 222

(d) start: 100
inTheAddSomeMethod: 100

(e) start: 100
inTheAddSomeMethod: 222

Question 14

[2 marks] The following method takes an array as input and returns a new array with the values in the reverse order:

```
public int[] reverse (int[] values) {  
    int[] result = new int[values.length];  
    for (int i = 0; i < result.length; i++) {  
        Missing statement  
    }  
    return result;  
}
```

In order for the `reverse ()` method to execute correctly the '**Missing statement**' line of code should be replaced by which of the following statements?

- (a) `result[i] = values[values.length-1];`
- (b) `result[values.length-i-1] = result[i];`
- (c) `result[i] = values[values.length-i-1];`
- (d) `result[i] = values[values.length+1];`
- (e) `result[values.length-1] = values[i];`

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

SECTION B

Question 15 (12 marks)

Complete the method header for the following three methods (i.e., complete the first line of the method definition).

a) The method, `changeArray()`, is called in the following way:

```
int[] values = {1,2,3,4,5};  
changeArray(values, values[0], true);
```

```
private void changeArray(int[] numbers, int value, boolean add) {  
    if (add) {  
        numbers[1] = numbers[1] + value;  
    } else {  
        numbers[1] = numbers[1] - value;  
    }  
}
```

(4 marks)

b) The method, `isPrime()`, is called in the following way:

```
if (isPrime(100)) {  
    System.out.println("100 is a prime number");  
}
```

```
private boolean isPrime(int num) {  
    int upTo = (int)Math.sqrt(num);  
    int i = 2;  
    while (i <= upTo) {  
        if (num % i == 0) {  
            return false;  
        }  
        i++;  
    }  
    return true;  
}
```

(4 marks)

c) The method, `replaceCharacterAtPosition()`, is called in the following way:

```
String myString = replaceCharacterAtPosition(0, "abc", 'X');
```

```
private String replaceCharacterAtPosition(int position,
                                         String word, char replacement) {

    String before = word.substring(0, position);
    String after = word.substring(position + 1);
    return before + replacement + after;
}
```

(4 marks)

Question 16 (12 marks)

A factory requires a program to calculate the number of containers needed to store a given number of items. Each container can fit up to 10 items.

- a) In this exercise, you need to complete the `containersNeeded()` method which is passed the number of items to store as a parameter. The method should calculate and return the number of containers sufficient to store that many items.

For example, if there are 10 items to store, then only 1 container is needed. However, if there are 11 items to store, then 2 containers are needed (one container will be full, and the other will have just one item).

```
private int containersNeeded(int items) {
```

```
    int number = items / 10;

    if (items % 10 > 0) {
        number++;
    }

    return number;
}
```

(8 marks)

```
}
```

- b) The factory would like to determine how many containers are needed for 123 items. They want to use your `containersNeeded()` method. Complete the `start()` method below showing how the `containersNeeded()` method should be called. The `start()` method should calculate the number of containers needed to store 123 items and then print this result to the screen.

```
public void start() {
```

```
    int result = containersNeeded(123);

    System.out.println(result + " containers");
}
```

(4 marks)

```
}
```

Question 17 (10 marks)

Consider the following method definition:

```
private String change(String word) {  
    if (word.length() < 3) {  
        return word;  
    } else {  
        int pos = word.length() / 2;  
        String first = word.substring(0, pos);  
        String last = word.substring(pos, word.length());  
        return last + first;  
    }  
}
```

Complete the output produced by the following code:

```
System.out.println("1. " + change("hotdog"));  
System.out.println("2. " + change("abc"));  
System.out.println("3. " + change("12"));  
System.out.println("4. " + change("computers"));
```

1: **doghot**
2: **bca**
3: **12**
4: **uterscomp**

(10 marks)

Question 18 (15 marks)

a) Complete the output produced by the following section of code.

```
System.out.print("i: ");  
  
for (int i = 7; i < 20; i = i + 5) {  
    System.out.print(i + " ");  
}
```

i: 7 12 17

(4 marks)

b) Complete the output produced by the following section of code.

```
int num = 20;  
System.out.print("num: ");  
  
while (num > 2) {  
    num = num - 5;  
    System.out.print(num + " ");  
}
```

num: 15 10 5 0

(4 marks)

- c) Complete the `getSumEvens()` method which takes two `int` parameters: the starting number of the sum and the last number to be added to the sum. The method returns the sum of all the even numbers between the start number and the last number. **You can assume that both parameters are even numbers.** For example, if the following code is run with the completed `getSumEvens()` method:

```
int sum = getSumEvens(4, 12);  
System.out.println("4 + 6 + 8 + 10 + 12 = " + sum);  
  
sum = getSumEvens(10, 30);  
System.out.println("10 + 12 + 14 + 16 + ... + 30 = " + sum);
```

the output is:

```
4 + 6 + 8 + 10 + 12 = 40  
10 + 12 + 14 + 16 + ... + 30 = 220
```

```
private int getSumEvens(int start, int last) {
```

```
    int sum = 0;  
    //assumes start and last are even numbers  
  
    for (int i = start; i <= last; i++) {  
        sum = sum + i;  
    }  
  
    return sum;
```

```
}
```

(7 marks)

Question 19 (8 marks)

Complete the program below which calculates and displays the cost of movie coupons. The program prompts the user for the number of coupons required, calculates the cost of the coupons and displays the cost. Each coupon costs \$8. Two example outputs using the completed program are shown below. (The user input is shown in bold and in a larger font.)

```
> java CouponsApp
*** BUY MOVIE COUPONS ***
```

```
Number required: 3
Cost: $24
```

```
> java CouponsApp
*** BUY MOVIE COUPONS ***
```

```
Number required: 12
Cost: $96
```

Note: You can assume that the Keyboard class is available in the same folder as MovieCouponsProgram.

```
public class MovieCouponsProgram {
```

```
    public void start() {
```

```
        final int COST_PER_COUPON = 8;
```

```
        System.out.println("*** BUY MOVIE COUPONS ***");
```

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

```
        int cost, numberRequired;
```

```
        System.out.print("Number required: ");
```

```
        String input = Keyboard.readInput();
```

```
        numberRequired = Integer.parseInt(input);
```

```
        cost = numberRequired * COST_PER_COUPON;
```

```
        System.out.println("Cost: $" + cost);
```

```
    }
```

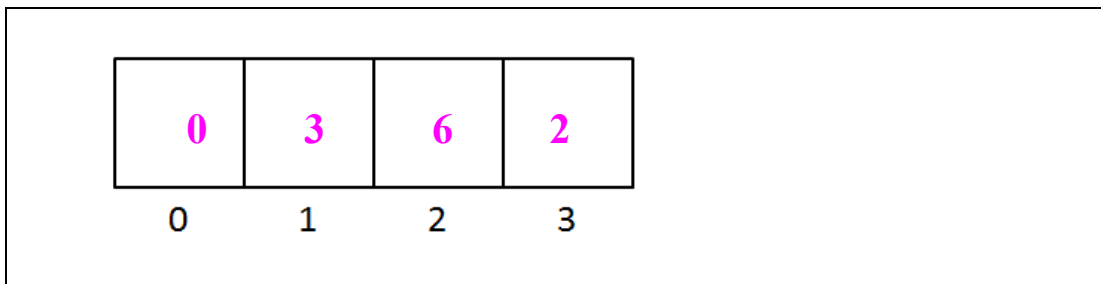
```
}
```

(8 marks)

Question 20 (15 marks)

- a) In the boxes below, show each element of the `numbers` array after the following code has been executed.

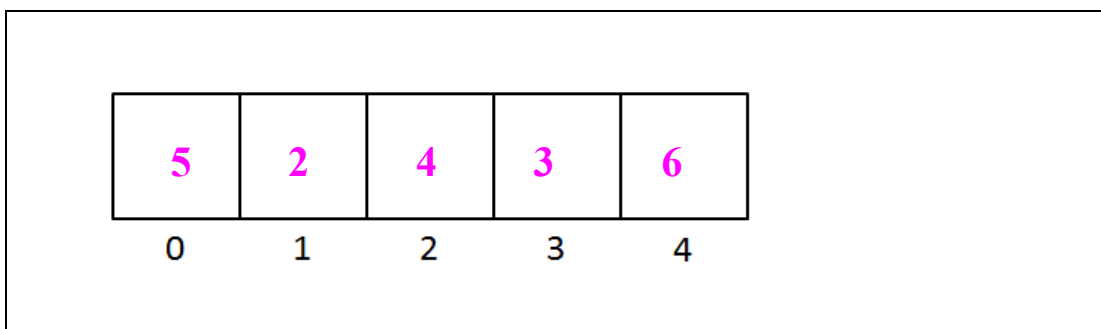
```
int[] numbers = {0, 3, 2, 1};  
numbers[3] = numbers[1] - numbers[3];  
numbers[numbers[3]] = 6;
```



(4 marks)

- b) In the boxes below, show each element of the `numbers` array after the following code has been executed.

```
int[] numbers = {5, 2, 3, 4, 6};  
int pos1 = numbers[1];  
int pos2 = numbers.length - 2;  
int temp = numbers[pos1];  
numbers[pos1] = numbers[pos2];  
numbers[pos2] = temp;
```



(4 marks)

- c) Complete the `getNumberOfEvens()` method which takes an `int` array as a parameter. The method returns the number of even values in the array. For example, if the following code is run with the completed `getNumberOfEvens()` method:

```
int[] nums1 = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
int numEvens = getNumberOfEvens(nums1);  
System.out.println(numEvens + " even numbers");  
  
int[] nums2 = {4, 6, 2, 8, 0, 5, 7};  
numEvens = getNumberOfEvens(nums2);  
System.out.println(numEvens + " even numbers");
```

the output is:

```
4 even numbers  
5 even numbers
```

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

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

```
}
```

(7 marks)