

ICT131

Examination – January Semester 2016

Introductory Programming and Object Oriented Concepts Using Java

Tuesday, 17 May 2016

1:00 pm – 3:00 pm

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS:

1. This examination contains **FOUR (4)** questions and comprises **NINE (9)** printed pages (including cover page and appendix A).
2. You must answer **ALL** questions.
3. This is a Closed Book examination.
4. All answers must be written in the answer book.
5. Appendix A contains some Java API which you might need.

At the end of the examination

Please ensure that you have written your examination number on each answer book used.

Failure to do so will mean that your work cannot be identified.

If you have used more than one answer book, please tie them together with the string provided.

**THE UNIVERSITY RESERVES THE RIGHT NOT TO MARK YOUR
SCRIPT IF YOU FAIL TO FOLLOW THESE INSTRUCTIONS.**

Answer all the questions. (Total 100 marks)

Question 1

- (a) The shape in Figure Q1(a) is an ellipsoid.

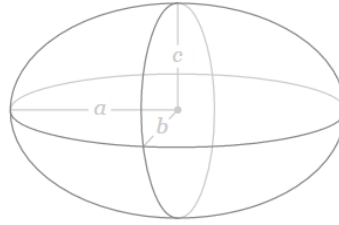


Figure Q1(a)

Suppose the radii of the ellipses about the x -, y - and z -axis are a , b and c respectively, then the formula to calculate its volume is

$$\text{Volume} = \frac{4}{3} \pi abc$$

where π is defined in the Math class in the Java library and a , b and c need not be whole numbers.

Using the Java-supplied classes and reference to the Java documentation provided in appendix, write a Java program that

- accepts three input arguments from the command line and converts them decimal numbers for a , b and c ,
- calculates the volume with the given formula,
- displays the volume of the ellipsoid together with the values of a , b and c , using `printf`. Each number is to be displayed with 2 decimal places as shown below, i.e. sample inputs of a , b and c equals to 2.1, 1, 1.1 respectively would return:

Volume = 9.68 given $a = 2.10$, $b = 1.00$, $c = 1.10$

Note: You MUST NOT hardcode the values for a , b and c in the formula.

(7 marks)

- (b) Examine the program in Figure Q1(b) and outline the output of the program.

```
1 public class Q1b {
2     public static void main (String [] args){
3         int [] n = { 1, 2, 3, 4, 5, 6};
4         methodX(n);
5         System.out.println("(main)1 " + "n[1] = " + n[1]+
6             " n[3]= " + n[3]);
7         methodX(n[2], n[5]);
8         System.out.println("(main)2 " + "n[1] = " + n[1]+
9             " n[3]= " + n[3]);
10        n = methodX(n[3], n[5], n[0], n[2]);
11        System.out.println("(main)3 " + "n[1] = " + n[1]+
12            " n[3]= " + n[3]);
13        methodX(n[1]);
14        System.out.println("(main)4 " + "n[1] = " + n[1]+
15            " n[3]= " + n[3]);
16        n[3] = methodX(n[3]);
17        System.out.println("(main)5 " + "n[1] = " + n[1]+
18            " n[3]= " + n[3]);
19    }
20    public static void methodX (int a1, int a2){
21        int temp = a1;
22        a1 = a2;
23        a2 = temp;
24        System.out.println("(methodX)1 " + "a1 = " + a1 +
25            " a2 = " + a2);
26    }
27    public static void methodX( int [] a){
28        for (int i = 0; i < a.length-1; i++)
29            a[i] = a[i+1];
30        for (int i = 0; i < a.length; i++)
31            System.out.println("(methodX)2 a[" + i + "] = " +
32                a[i] + " ");
33    }
34    public static int [] methodX (int a1, int a2, int a3,
35        int a4){
36        int [] a = {a1, a2, a3, a4};
37        for (int i = 0; i < a.length; i++)
38            System.out.println("(methodX)3 a[" + i + "] = " +
39                a[i] + " ");
40        return a;
41    }
42    public static int methodX (int a){
43        System.out.println("(methodX)4 a = " + (a+=3));
44        return a;
45    }
46 }
```

Figure Q1(b)

(18 marks)

Question 2

Outline the control structures of structured programming. For each of the following scenarios in part (a) to (d),

- **without repeating your choice for the other scenarios, specify with reasons,** the appropriate control structure(s) to use from the following list:

if if-else switch for while do-while

- **develop a program segment** to show how the chosen structure is applied to the scenario.

- (a) A person's age has been read into the variable `age`.

Write a program segment to determine whether the person is a child (12 years and below), a teenager (13 to 19), or an adult (20 and above) and display the category the person belongs to.

(6 marks)

- (b) Eligible students can choose to study a third language. Assume that the language code has been read into the variable `code`.

Write a program segment to help prevent incorrect language code entry. For the code values 6, 7, 8, and 65, display the language selected as French, Japanese, German and Spanish respectively. Print `Invalid` for any other code.

(6 marks)

- (c) The number of lines to print the word `Hello` has been read into the variable `num`.

Write a program segment to print the word `Hello` as many times as the value in `num`.

(6 marks)

- (d) The number of traffic offense tickets a parking warden will issue in a day varies.

Write a program segment that whenever the parking warden enters a vehicle number, your program segment must read the offense type which is a whole number and print a ticket to display the vehicle number and offense type. The traffic warden enters `end` for vehicle number to terminate data entry. It is possible that there is no ticket issued on a day.

You may assume that a `Scanner` object has been created and is referenced with the variable `sc`.

(7 marks)

Question 3

Develop a Java program using the control structures of structured programming on an array data structure for an application that uses one array `names` to record student names and another array `scores` to record student test scores. A test score is a whole number. The arrays can record a maximum of 25 data each.

Write a complete Java program with the following methods:

- (a) A method `recordANewScore` that accepts the two arrays and an integer `count` which is the number of data already recorded in the arrays.

If `count` is already the maximum, the method prints the error message: `No more scores can be added!` Otherwise, the method prompts the user for a test score and a name, records them in the appropriate arrays and increments `count` as one more data is recorded.

This method returns the value of `count`.

(6 marks)

- (b) A method `listFailureCases` that accepts the two arrays and an integer `count` which is the number of data already recorded in the arrays.

If no data has been recorded in the arrays, the method displays the message: `No data recorded yet!` Otherwise, the method displays the list of names together with the scores of students for students with scores below 40.

The method also prints the average test score for the failure cases if there are failures. But if there are no failure cases, the method prints: `There is no failure case.`

(10 marks)

- (c) A method `displayMenu` that displays the following menu:

```
Menu
1. Enter new test scores
2. List failure cases
0. Exit
```

(2 marks)

- (d) A method `main` which does the following:

- declare and create two arrays `scores` and `names` that can record a maximum of 25 test scores and 25 names of students respectively
- declare and initialise `count`, the number of data already recorded in the arrays to zero,
- repeatedly display a menu to allow the user to select an option and call the appropriate method to perform the task according to his selection until the user chooses to exit the program.

(7 marks)

Question 4

CupACoffee is an upmarket restaurant serving gourmet coffee. To encourage regular customers, it plans to introduce a member cash card program to allow customers to earn bonus points when they top up the cash card. Bonus points can be used to redeem finger bites. You are tasked to develop a prototype for CupACoffee using object-oriented programming approach to allow it to implement the member cash card program.

Each member cash card has a

- card number which consists of 10 digits and a character which acts as a checksum character,
- type of membership – S for secondary student, T for tertiary student and A for adult
- cash balance,
- current bonus point balance,

Write a Java class that models a member cash card. This class should have the following:

- (a) Suitable instance variables for each piece of information given above. (4 marks)
- (b) A constructor with the appropriate actions on the instance variables. The default value for current bonus point balance is 0. (2 marks)
- (c) The get and set methods for the instance variable for type of membership, and only the get method for current bonus point balance and card number. (4 marks)
- (d) A `topUp()` method that accepts an amount to add to the cash balance. For every \$10 top up, 1 bonus point is added. In addition, a secondary school or tertiary member gets 1 additional bonus point for every \$20 top up. (5 marks)
- (e) A `redeem()` method that decrements the bonus point balance by 5 and returns true if the redemption is successful. The method does not decrement and returns false if there is currently fewer than 5 bonus points. (4 marks)
- (f) A `purchase()` method that accepts an amount to be subtracted from the cash balance. If the cash balance is sufficient, subtract the amount from the cash balance and return true. If the cash balance is less than the purchase, do not subtract from the cash balance but simply return false. (4 marks)
- (g) A `toString()` method that returns a string containing the values of the instance variables with descriptions. (2 marks)

----- END OF PAPER -----

Appendix A

Class Double

Modifier and Type	Method and Description
double	<u>doubleValue()</u> Returns the double value of this Double object.
static double	<u>parseDouble(String s)</u> Returns a new double initialized to the value represented by the specified String, as performed by the valueOf method of class Double.

Class Integer

Modifier and Type	Method and Description
int	<u>intValue()</u> Returns the value of this Integer as an int.
static int	<u>parseInt(String s)</u> Parses the string argument as a signed decimal integer.
static <u>Integer</u>	<u>valueOf(int i)</u> Returns an Integer instance representing the specified int value.

Class Math

Modifier and Type	Field and Description
static double	<u>E</u> The double value that is closer than any other to e, the base of the natural logarithms.
static double	<u>PI</u> The double value that is closer than any other to π , the ratio of the circumference of a circle to its diameter.

Class Scanner

Modifier and Type	Method and Description
boolean	<code>hasNextBoolean()</code> Returns true if the next token in this scanner's input can be interpreted as a boolean value using a case insensitive pattern created from the string "true false".
boolean	<code>hasNextDouble()</code> Returns true if the next token in this scanner's input can be interpreted as a double value using the <code>nextDouble()</code> method.
boolean	<code>hasNextInt()</code> Returns true if the next token in this scanner's input can be interpreted as an int value in the default radix using the <code>nextInt()</code> method.
String	<code>next()</code> Finds and returns the next complete token from this scanner.
boolean	<code>nextBoolean()</code> Scans the next token of the input into a boolean value and returns that value.
double	<code>nextDouble()</code> Scans the next token of the input as a double.
int	<code>nextInt()</code> Scans the next token of the input as an int.
String	<code>nextLine()</code> Advances this scanner past the current line and returns the input that was skipped.

Class String

Modifier and Type	Method and Description
char	<u>charAt</u> (int index) Returns the char value at the specified index.
int	<u>compareTo</u> (String anotherString) Compares two strings lexicographically.
int	<u>compareToIgnoreCase</u> (String str) Compares two strings lexicographically, ignoring case differences.
<u>String</u>	<u>concat</u> (String str) Concatenates the specified string to the end of this string.
int	<u>indexOf</u> (String str) Returns the index within this string of the first occurrence of the specified substring.
int	<u>indexOf</u> (String str, int fromIndex) Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.
int	<u>lastIndexOf</u> (String str) Returns the index within this string of the last occurrence of the specified substring
int	<u>length</u> () Returns the length of this string.