香港公開大學
THE OPEN UNIVERSITY
OF HONG KONG

COMPS202F
Marker No.:
Total Mark:

2020 Autumn Examination (UG)

JAVA PROGRAMMING FUNDAMENTALS (2020 Autumn Term)

18 Dec 2020 **Time Allowed: 2 hours** **14:00–16:00**

Student Number								

THIS IS AN ONLINE OPEN-BOOK EXAMINATION

Instructions:

1. This examination paper is available on OLE Online Exam Paper(s) page.
2. You should answer the examination paper using **your OWN efforts only**. Any plagiarism behaviour discovered will have serious consequences.
3. This is an open-book examination. You can read books and access the Internet. However, you cannot copy answers from any source.
4. Use your own paper to answer the questions of this examination paper in English. You may optionally print our answer book (on OLE) for writing your answers. **Write your student ID and name on your answer sheets.**
5. You can start as soon as you successfully download the question paper.
6. Read the instructions in the examination paper carefully and **write** the question numbers and answers clearly. It may not be possible to award marks where the writing is very difficult to read. **You are required to WRITE your answers and typed answers are NOT allowed** (unless prior approval is given)**.**
7. At the end of the examination, scan your answers in **black and white**. Check each page to **make sure each image is not out of focus and sharp enough to be seen.** Generate and upload a PDF file containing your answers to OLE “Online Exam”.
8. As a backup, attach your PDF file in an email to kwlee@**study**.ouhk.edu.hk with subject “COMPS202F Exam: 11223344” (where 11223344 is your student ID). Remember to CC your email address a copy.

   Another backup email address is comps202f@live.ouhk.edu.hk, If either (1) OLE upload or (2) sending email to the first email address is not successful, send a copy of your answers to this email address. If both (1) and (2) are fine, this one is not needed.

PART I (60 marks)

(i) *You should attempt* **ALL** *the questions in this part of the examination paper.*

(ii) *There are altogether* ***eight*** *questions in this part.*

(iii) *You are advised to spend one hour and ten minutes on this part.*

Question 1

(a) The following class contains some errors. After removing them, the `main()` method can be run directly by the Java virtual machine. Suggest two simple corrections to remove the errors.

```
public class Q1PartA<> {
  public static void main(Double[] args) {
    System.out.println("Correct errors.");
  }
}
```

[4 marks]

(b) Write a class `Camera` with two private attributes: `brand` (of type `String`) and `price` (of type `double`).

[3 marks]

Question 2

The following method is written to find and print the product of the numbers: -1, -2 and -3:

```
public void aMethod() {
  int from = -1;
  int to = -3;
  int product = 1;
  for (int i=from; i<-3; i++) {
    product *= i;
  }
  System.out.println("The product is "+product);
}
```

(a) State the correct value of the product. Also give the value(s) printed by the above method.

[3 marks]

(b) Suggest at most two corrections to remove the bugs.

[4 marks]

Question 3

A class Circle has an attribute diameter (of type double).

(a) Write the getter and setter methods of the attribute.

[3 marks]

(b) Write a constructor with a single parameter which initializes the attribute using the parameter.

[1 mark]

(c) What is the output of the following program segment?

```
double diameter0 = 20;

double diameter1 = 40;

Circle circle2 = new Circle(60);

Circle circle3 = new Circle(80);

diameter1 = 100;

circle2 = circle3;

circle3.setDiameter(diameter1);

System.out.println(diameter0+ ", " + circle2.getDiameter());
```

[3 marks]

Question 4

(a) A class Scanner in Java can be used to get user input and its methods can be used after the following statement is executed:

```
Scanner input = new Scanner(System.in);
```

One of its methods nextDouble() returns the next double value from the keyboard. Now you are requested to write a method enterDiameter() which displays the message "Input diameter: " and uses the method input.nextDouble() to get the user input. The process should be repeated until the input is non-negative. Finally the input should be returned. You can assume the aforementioned statement starting with Scanner has already been executed.

[4 marks]

(b) Write a method category(double diameter) which returns a suitable type in string form according to the following:

Condition	Type
diameter >= 60	Large
20 <= diameter < 60	Medium
diameter < 20	Small

[3 marks]

Question 5

A method `secret()` of a class `F` is shown below:

```
public int secret(String[] array, String string) {
  int num = 0;
  for (int i=0; i<array.length; i++)
    if (array[i].compareTo(string) >= 0)
      num++;
    else
      num--;
  return num;
}
```

A program segment using the method is:

```
F f = new F();
String[] stringA = {"Ada", "Bill", "Issac", "aaron", "Homantin"};
System.out.println(f.secret(stringA, "Billy"));
System.out.println(f.secret(stringA, "Issac"));
String[] stringB = {};
System.out.println(f.secret(stringB, "Ada"));
```

(a) Write down the output when the above segment is executed.

[3 marks]

(b) Explain, in a paragraph, what is the purpose of the method `secret()`.

[5 marks]

Question 6

Write a class method `printY(int width)`, which belongs to a class `Figure`, to print a figure using asterisks similar to the figure below (where "□" denotes a space and `width` is an odd integer).

The figure is generated by `Figure.printY(7)`.

```
*□□□□□*
□*□□□*□
□□*□*□□
□□□*□□□
□□□*□□□
□□□*□□□
□□□*□□□
```

[8 marks]

Question 7

(a) Declare a local `String` variable `id` and initialize it with your 8-digit student ID in string form. Also declare another local `String` variable `surname` and initialize it with your surname. What is the output of the following program segment?

```
String str = '\t' + id + surname;
System.out.println(str.indexOf("3") + " : " + str.substring(9) + "\n"
                   + str.trim().substring(5,8)+ "#");
```

[3 marks]

(b) Write a method `average(ArrayList<Integer> numList)` which returns the average value of the integers which are less than 40 and stored in `numList`, where `numList` contains at least one integer less than 40. Choose a suitable return type.

[5 marks]

Question 8

Given the following class:

```
public class Item {
  private String id;
  private double weight;
  public double getWeight() { return weight; }
  public void setWeight(double aWeight) { weight = aWeight; }
}
```

Each `Item` object has a unique id. Using an **enhanced for** loop, write a method `heavyItems(Item[] item, double threshold)`. The method returns a **set** of `Item` objects in which the `weight` of each is not less than `threshold`.

[8 marks]

[END OF PART I]

PART II (40 marks)

(i) You should attempt **ALL** *questions. Each question is worth 20 marks.*
(ii) Show all your work steps.
(iii) You are advised to spend fifty minutes on this part.

Question 9

Some information of a computer shop is shown in the following table:

CPU Model No.	Price($)
Ryzen 3	780
Ryzen 5	1160
Core i3	930

(a) Write a class `CPU` with attributes `modelNumber` and `price`. Add a constructor `CPU(String modelNumber, double price)` to initialize the two attributes with the parameters.

[3 marks]

(b) Write another class `ComputerShop` with one single attribute `cpuList` of type `ArrayList<CPU>` and initialize the attribute with a new object in the declaration. Add any `import` statement(s) needed.

[3 marks]

In (c) and (d), you can assume the getter and setter methods of the attributes of `CPU` have been written.

(c) Write a method `findCPU(String aModelNumber)` of `ComputerShop` which returns the **`CPU` object** with CPU model number `aModelNumber`. You can assume there is at most one such object. If no matching CPU can be found, please return `null`.

[6 marks]

(d) Write a method `lowPriceCPU(double bound)` of `ComputerShop` to return a **`HashSet` of `CPU` objects** with `price` not more than `bound`.

[8 marks]

Question 10

To perform simple encryption, a lower case letter can be rotated by `k` positions. For example, if `k` is 2, 'a' becomes 'c', 'b' becomes 'd' and so on. The last letters need to be rotated back. 'y' becomes 'a' and 'z' becomes 'b'. Uppercase letters can be rotated in the same way. Other characters are unchanged.

(a) Write a method `rotate(String str, int k)` which returns the rotated string.

[5 marks]

(b) To make the encryption more sophisticated, we can use a dynamic value for `k`. One choice is the highest no. of occurrences of letters in the first `n` characters. Write a method `findHighest(String str, int n)` to return the highest no. of occurrences of the letters in the first `n` characters in the string `str`. For example, `findHighest("abcabbbc", 6)` returns 3 because "**`b`**" occurs 3 times in the first 6 characters.

[7 marks]

(c) Using the methods written, write a method `encrypt(String str, int n)` to return an encrypted string by rotating letters by k positions, where k is the highest no. of occurrences of letters in the first `n` characters in `str`.

[2 marks]

(d) Using the method(s) written, write a method `decrypt(String str, int n)` which decrypts the encrypted string `str`, which was encrypted by the method in part (c), and returns the original string. The parameter `n` has its usual meaning as in parts (b) and (c).

[6 marks]

Appendix: Concise Java Statement Examples and Partial Method List

This appendix is provided to reduce the load of memorizing the syntax and methods learnt. This is not a complete reference and total correctness is not guaranteed. Some methods not listed here need to be used.

Statement Examples

```
int a, b, c, e;
int[] f = new int[9];
double d;
TicketCounter tc = new TicketCount();
if (a == b && b != 1 || c <= 0) {
  d = 0;
} else {
  d = 1;
}
switch (e+2) {
  case 2 : case 5:
     f[4] = (int) d; break;
  default :
     tc.increase(); break;
}
```

```
public class Example {
  boolean good;
  int j=0, k=0, m=0, n=0;
  public double loops(String s) {
     for (int i=0; i<n; i++) {
        j += i;
     }
     do {
        k = k + 2;
     } while ( k < 5 );
     while (true || k > 2) {
        m++;
     }
     return (double) k;
  }
}
```

Method List

```
Collection --    add(o), contains(o), clear(), remove(o), size()
  List --           add(i,o), get(i), indexOf(o), remove(i), set(i,o)
  Set --            <see Collection>


String --        charAt(i), compareTo(s), equals(o), indexOf(c),
                 indexOf(s), length(), replace(c,c), substring(i,j),
                 toCharArray(), toLowerCase(), toUpperCase()
```

where `c:char, i,j: int, o: Object, s: String`

[END OF EXAMINATION PAPER]