Beijing-Dublin International College

AUTUMN TRIMESTER RESIT EXAMINATION - (2022/2023)

School of Computer Science

COMP2011J Object Oriented Programming

Dr. Robert Ross
Assoc. Prof. Neil Hurley
Dr. Seán Russell*

Time Allowed: 90 minutes

Instructions for Candidates:

Answer **All** questions.

BJUT Student ID:____________________ **UCD Student ID:**____________________

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge:____ **(Signature)**

Instructions for Invigilators

Non-programmable calculators are permitted.

No rough-work paper is to be provided for candidates.

Question 1:

a. What is the biggest difference between object-oriented programming and procedural programming? Compare how data is accessed by functions in procedural programs and by methods in object-oriented programs. What effect does this have on the number of parameters required for methods? **(10 points)**

b. Explain the difference between an expression and a statement. Name and explain the use of one of the statements in Java you have learned, give an example showing the statement being used. Can a piece of code be both a statement and an expression? Give an example. **(10 points)**

c. What is the implicit parameter? Given the following example of comparing two date objects (`birthday` and `today`), which is the implicit parameter?

```
today.sameDate(birthday);
```

Rewrite the following method to show the implicit parameter in the code using the keyword `this`:

```
public void summate(int a){
  count = count + a;
  if (count > 28){
    return sum();
  }
}
```

(10 points)

d. Explain in your own words the OOP concept **cohesion**. Describe *high* and *low* cohesion. Which is considered good? Why? **(10 points)**

e. Explain the difference between the following pieces of code (Assuming `s` is a `String` variable). Will they **always** return the same result? Why?

```
s == "Hello"
```

(a)

```
s.equals("Hello")
```

(b)

(10 points)

(Question Total 50 points)

Question 2:

a. What effect does the keyword `final` have on a variable? Explain the difference between a final primitive variable and a final object variable.

(10 points)

b. Describe how Interfaces can be used to enable code reuse in Java. Describe in detail how an interface would be defined and implemented to make reusable code. **(10 points)**

c. Explain the phrase “prefer composition over inheritance” in your own words. How is this achieved in a Java class? Give one benefit and one drawback of choosing to create a class using composition instead of inheritance. **(10 points)**

d. What is a lambda function? When can we use a lambda function in Java? Define a lambda function to implement the `Calculator` interface defined below. The lambda function should calculate the product of these numbers multiplied by 2 (i.e $a * b * 2$). What parts of the syntax of a lambda function are optional?

```java
interface Calculator {
    int performCalculation(int a, int b);
}
```

(10 points)

e. Explain the concept of inversion of control. What is required in our code to make this work? Why is this necessary when we are programming a graphical user interface?

(10 points)

(Question Total 50 points)

Question 3:

a. What is a Stack Trace? Explain the order that a stack trace is is printed in and why.
(10 points)

b. What is the difference between the information in a text file and a data file? Give an example of both storing the integer value 123. What would happen if I try to use `readLine` from a `BufferedReader` to read information from a data file containing integer values?
(10 points)

c. Describe the difference between traditional testing and unit testing. How do we know when we have enough code coverage in our testing?
(10 points)

d. What is the purpose of code documentation (e.g. Javadoc)? Write a suitable javadoc comment for the following method. Parameters, return types and exceptions should be specified correctly (you can list yourself as the author).

```java
public int convertToInteger(String s) {
    if (s == null || s.length() == 0) {
        throw new IllegalArgumentException("Input is not valid!");
    }
    int result = 0;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) < '0' || s.charAt(i) > '9') {
            throw new NumberFormatException("Input is not a number!");
        }
        result = result * 10 + (s.charAt(i) - '0');
    }
    return result;
}
```

(10 points)

e. Explain how generics improves type checking in the compiler. Give an example of a situation where an error would not be detected without generics. **(10 points)**

(Question Total 50 points)

(Exam Total 150 points)