قوليتيكنيك بروني
POLITEKNIK BRUNEI

END-OF-SEMESTER EXAMINATION (RESIT)

SEMESTER 2, 2022-23

SCHOOL OF INFORMATION & COMMUNICATION TECHNOLOGY

NS4307

NETWORK PROGRAMMING

TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES:

- **DO NOT TURN OVER THE PAGE UNTIL YOU ARE TOLD TO DO SO.**
- Write down the details required on the front cover of the answer booklet provided. Make sure you complete the details on any additional answer booklets that you use.
- Answer ALL questions.
- The total mark for this paper is **70**. The number of marks for each question or part question is shown in brackets.
- Your answer booklets and any other paper used in the examination must be handed in to the invigilators before leaving the room or hall. You may take your own examination paper with you unless otherwise instructed.
- Write in **dark blue** or **black** ink pen.

MATERIALS SUPPLIED:

Answer booklet (1).

MATERIALS PERMITTED:

Calculator.

THIS PAPER CONSISTS OF 8 PRINTED PAGES (INCLUDING THE COVER PAGE).

STRUCTURED QUESTIONS [TOTAL MARKS: 70]

Answer **ALL** questions.

QUESTION 1 [Total marks: 14]

a) "Version control is a system that records changes to a file or set of files over time." Explain why that is the case. **[1 mark]**

b) There are three (3) types of Version Control Systems (VCS) and one of them is Centralised Version Control Systems. List the other **two (2)** types of VCS. **[1 mark]**

c) Illustrate the basic structure of Centralised Version Control Systems. **[4 marks]**

d) State any **three (3)** typical features in Version Control Systems. **[3 marks]**

e) Explain what happens when you clone a Git repository. **[2 marks]**

f) State **and** explain **two (2)** ways of using Git to update the development environment. **[3 marks]**

QUESTION 2 [Total marks: 14]

a) Explain in detail the difference between Java Binary Input / Output **and** Java Text Input / Output in terms of writing and reading to/from a file. (Note: You may illustrate using diagram or use an example to further enhance your explanation) **[4 marks]**

b) Your colleague has just started to learn Java Binary Input / Output and tried to implement a console application that read/write primitive and String data types only from/to a file in a Windows operating system. Currently, your colleague needs your help to complete the implementation where it is commented with TODO.

The following Table 1 is a Java class called ChildrenDetails.java which your colleague is currently implementing.

```
ChildrenDetails.java
1.   public class ChildrenDetails {
2.     public static void main(String[] args)
3.         throws FileNotFoundException, IOException {
4.       Scanner scanner = new Scanner(System.in);
5.       // The directory is valid and JaiChildren.dat already exists
6.       File file = new File("JaiChildren.dat");
7.       try(
8.         // TODO 1
9.       ) {
10.        System.out.println("Children Details in the file.");
11.        while(readData.available() != 0) {
12.          System.out.println("######################");
13.          System.out.println("Name: " + readData.readUTF());
14.          System.out.println("Year of Birth: " + readData.readInt());
15.          System.out.println("Month of Birth: " + readData.readInt());
16.          System.out.println("Day of Birth: " + readData.readInt());
17.          System.out.println("######################");
18.        }
19.      }
20.      try(
21.        // TODO 2
22.      ) {
23.        System.out.println("Filling in more children details:");
24.        System.out.println("Name:");
25.        String name = scanner.nextLine();
26.        System.out.println("Date of Birth (dd/mm/yyyy):");
27.        String[] dob = scanner.nextLine().split("/");
28.        int day = Integer.parseInt(dob[0]);
29.        int month = Integer.parseInt(dob[1]);
30.        int year = Integer.parseInt(dob[2]);
31.        // TODO 3
32.      }
33.      scanner.close();
34.    }
35.  }
```

Table 1

i. Based on Table 1, implement the appropriate InputStream object in Line 8, commented with TODO 1, that will read the File object, file, in Line 6. **[3 marks]**

[Turn over

ii. Based on Table 1, implement the appropriate OutputStream object in Line 21, commented with TODO 2, that will append the File object, file, in Line 6. **[3 marks]**

iii. Based on Table 1, implement the appropriate statements in Line 31 commented with TODO 3 that will write data in Line 25, Line 28, Line 29, and Line 30 once, using the OutputStream object created in Question 2.b) ii. to the File object, file, in Line 6 accordingly. The data written into the file must follow the order of it being read in Line 13 – Line 16 with its integrity still maintained. **[4 marks]**

QUESTION 3 [Total marks: 14]

a) Explain Threads in Java. **[2 marks]**

b) The following Table 2 and Table 3 are **two (2)** Java classes called StudentServer.java and StudentClient.java, respectively.

```
StudentServer.java
1.   public class StudentServer {
2.     public static void main(String[] args) throws Exception {
3.
4.     }
5.   }
```

Table 2

```
StudentClient.java
1.   public class StudentClient {
2.     public static void main(String[] args) throws Exception {
3.
4.     }
5.   }
```

Table 3

i. Table 2 is a server application. Complete the implementation of Table 2 based on the following:

When the server application starts, it will listen to port number 9991. The server application should accept connection from a client application, and it should be able to send to and receive Java primitive and String data types only from the client application. (Note: write only the code statements in proper order. You are not required to rewrite the whole class, handle any exceptions being thrown by any of the statements or import any of the classes.) **[4 marks]**

ii. Table 3 is a client application that connects to the server application (Table 2). Complete the implementation of Table 3 based on the following:

It will request a connection to the server application hosted in IP address 107.212.10.203. The client application should be able to send to and receive Java primitive and String data types only from the server application. (Note: write only the code statements in proper order. You are not required to rewrite the whole class, handle any exceptions being thrown by any of the statements or import any of the classes.) **[3 marks]**

[Turn over

iii. Implement the communications between the server application (Table 2) and the client application (Table 3). Add further implementations to Table 2 and Table 3 based on the following order (Note: State whether the implementation is for server application or client application):

1. The client application sends String data, “20FTT9991”, to the server application.
2. The server application sends String data, “Abu”, and integer data, 10, to the client application **[5 marks]**

QUESTION 4 [Total marks: 14]

a) Differentiate between Uniform Resource Locator (URL) **and** Uniform Resource Identifier (URI). **[4 marks]**

b) Briefly explain the concept of Model View Controller. **[1 mark]**

c) The following Table 4 is a Java class called MainController.java.

MainController.java

```java
@Controller
public class MainController {

}
```

Table 4

i. Based on Table 4, Implement a method to handle Uniform Resource Identifier (URI) “/” where it will respond with a literal String “NS4307 Network Programming”. (Note: You are not required to rewrite the whole class.) **[4 marks]**

ii. Based on Table 4, Implement a method to handle Uniform Resource Identifier (URI) “/{message}” where it will respond with a literal String of the path variable. (Note: You are not required to rewrite the whole class.) **[5 marks]**

QUESTION 5 [Total marks: 14]

a) Before a Spring Web Application with Hibernate can connect to a database management system, the web application needs to be configured first. One of the properties that can be configured is spring.jpa.hibernate.ddl-auto. Explain the **four (4)** values: create, create-drop, update and validate, that can be assigned to the property. **[4 marks]**

[Turn over

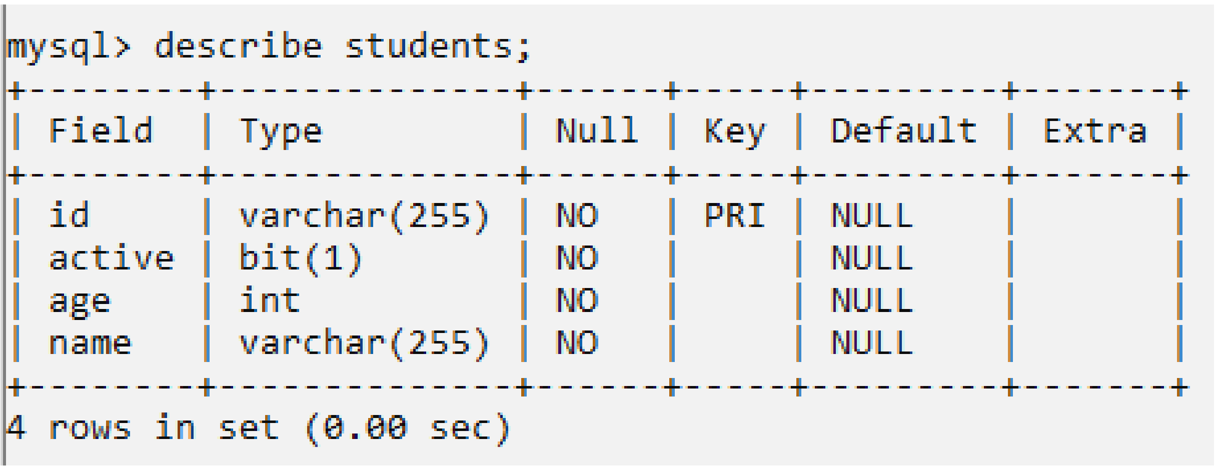
b) The following Table 5 is a Java class called Student.java and Table 6 is a command line (terminal) screenshot of a table details queried in MySQL Database Management System server, that are used for Spring Web Application with Hibernate.

Student.java

```
public class Student {

}
```

Table 5



```
mysql> describe students;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| id     | varchar(255) | NO   | PRI | NULL    |       |
| active | bit(1)       | NO   |     | NULL    |       |
| age    | int          | NO   |     | NULL    |       |
| name   | varchar(255) | NO   |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Table 6

Rewrite the Plain Old Java Object class, Student.java (Table 5), by making the class a Database entity that will be used by Hibernate to create the Database table (Table 6) with its field and constraints accordingly. **[10 marks]**

[END OF PAPER]