



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**FINAL EXAMINATION
SEMESTER II
SESSION 2021/2022**

- COURSE NAME : DOTNET PROGRAMMING
COURSE CODE : BIE 33103
PROGRAMME CODE : BIW / BIP
EXAMINATION DATE : JULY 2022
DURATION : 3 HOURS
INSTRUCTIONS : 1. ANSWER **ALL** QUESTIONS
2. THIS FINAL EXAMINATION IS AN **ONLINE** ASSESSMENT AND CONDUCTED VIA **CLOSE BOOK**
3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSE BOOK

THIS QUESTION PAPER CONSISTS OF **FIVE (5)** PAGES

Q1 (a) The Fibonacci sequence is a series of numbers (0, 1, 1, 2, 3,...) in which the next number is found by adding up the two numbers before it, for example the 2 is found by adding the two numbers before it (1+1), and the 3 by adding (1+2).

(i) Write a `<body></body>` code segment to produce the result shown in **Figure Q1(i)**.

Enter number of elements:

Figure Q1(i)

(2 marks)

(ii) Write the C# code behind segment to accept user input from the textbox as the amount of elements for the Fibonacci sequence, create a function to recursively derive the Fibonacci sequence up to the specified amount, and display the sequence after user clicks the “Generate Fibonacci” button based on **Figure Q1(i)**.

(5 marks)

(iii) Write a function in C# code behind to recursively obtain the nth Fibonacci element, with n representing the number that the user inputs using **Figure Q1(ii)** as a guide.

Get specific element in Fibonacci sequence:

Figure Q1(ii)

(3 marks)

(b) Write a C# code behind to produce a diamond shape made of the * symbol with the number of rows determined by user input, after the user clicks the “Diamond” button as in the **Figure Q1(iii)**. Consider the following:

- The textbox ID is Num_Rows
- The button ID is Diamond

Enter number of rows:

Figure Q1(iii)

(10 marks)



Q2 Write a complete C# code behind segment based on the description in **Figure Q2**.

Freddy Waffle Shop wants to create a web application to create bills for customer orders. The app produces a form that requests user to enter customer name and their order; if either the customer name or the order isn't specified, an error message will display. Customers can choose to order either chocolate, strawberry or kaya waffles, each of them costing RM2.50, and specify the amount of waffles. Afterwards, user will click the Calculate button, which displays the total price the customer needs to pay. If the total price is RM7.50 and above, there will be a 10% discount applied.

Figure Q2

(14 marks)

Q3 (a) Threads are units of processes that are responsible for the application code execution. By setting different execution paths or threads, complicated and time consuming operations can be multi-tasked or given priorities, with each thread performing a particular job.

(i) Write the output for the program as in **Figure Q3(i)**.

```
using System;
using System.Threading;

namespace Q3a_
{
    2 references
    public partial class Thread1 : System.Web.UI.Page
    {
        1 reference
        public void Proc_Thread(){
            for(int i = 0; i <= 6; i++){
                if (i % 2 == 0) {
                    Response.Write("This is even = " + i + "<br>");
                }

                else {
                    Response.Write("This is odd <br>");
                }
                Thread.Sleep(1000);
            }
        }
        0 references
        protected void Page_Load(object sender, EventArgs e){
            Thread thr1 = new Thread(Proc_Thread);
            thr1.Start();
            Response.Write("Start the count<br>");
            for (int a = 1; a < 5; a++)
            {
                Response.Write("Going On<br>");
            }
            thr1.Join();
        }
    }
}
```

Figure Q3(i)

(5 marks)

- (ii) Explain the process that occurs when the code is executed as in **Figure Q3(ii)**.

```

using System;
using System.Threading;
namespace Q3a_
{
    2 references
    public partial class Thread2 : System.Web.UI.Page
    {
        bool stopped = false;
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
            Thread thr1 = new Thread(new ThreadStart(() =>
            {
                while (!stopped){
                    Response.Write("Running..");
                    Thread.Sleep(1000);}
            }));
            Thread thr2 = new Thread(ProcThread2);
            thr1.Start();
            thr2.Start();
            thr2.IsBackground = true;
            thr1.Join();
        }

        1 reference
        public void ProcThread2(){
            Response.Write("I am number 1");
        }

        0 references
        protected void Button1_Click(object sender, EventArgs e)
        {
            stopped = true;
            Response.Write("stop");
        }
    }
}

```

Figure Q3(ii)

(5 marks)

- (b) Write the C# code segment to save a user's input into a database as in **Figure Q3(iii)**. Assume that **Table Q3(b)** has been created in the database.

The image shows a web form titled "DOTNET SURVEY" with the instruction "Fill in all the information fields below". The form contains the following fields:

- Name:** A text input field.
- Gender:** Two radio button options: Male and Female.
- Email:** A text input field.
- Phone Number:** A text input field.
- Submit:** A button at the bottom left of the form.

Figure Q3(iii)



Table Q3(b)

ColumnName	Data Type
name	varchar (30)
gender	tinyint (1)
email	varchar (50)
phone_no	varchar (15)

(20 marks)

Q4 Write ASP code segment in the <form> </form> to produce the output in Figure Q4. Consider the following information:

- The list for room types are Regular, Family, Luxury, and Suite
- For the validation expression for feedback ID, the total characters allowed for it is six, with the first two characters being a letter between a to z, and the rest are integers (0-9)
- Use the validation expression "\w+ ([-+. '] \w+) * @ \w+ ([-.] \w+) * \. \w+ ([-.] \w+) *" to validate email address pattern
- Use the validation expression "\ (^ [0 - 9] { 1 0 } \$) | (^ \ + [0 - 9] { 2 } \ s + [0 - 9] { 2 } [0 - 9] { 8 } \$) | (^ [0 - 9] { 3 } - [0 - 9] { 4 } - [0 - 9] { 4 } \$)" to validate phone number

Customer Feedback:

Feedback ID: Must enter feedback ID Invalid feedback ID

Room Type:

E-mail: Must enter e-mail Invalid e-mail format

Phone Number: Must enter phone number Invalid phone number

Customer Satisfaction: Excellent
 Good
 Bad
 Horrible

Figure Q4

(16 marks)

- END OF QUESTIONS -