

## ข้อสอบกลางภาค

วิชา MDT 211 Computer programming for Media 2

### ข้อกำหนด

1. ระยะเวลาการทำข้อสอบวันเสาร์ที่ 2 ตุลาคม พ.ศ.2564 เวลา 10.00 น. - วันอาทิตย์ที่ 3 ตุลาคม พ.ศ. 2564 เวลา 10.00 น.
2. อนุญาตให้ผู้สอบสามารถสืบค้น Internet และหนังสือได้
3. อนุญาตให้สามารถปรึกษา สนทนาแนวคิด ได้ตามความเหมาะสมโดยที่ Code ห้ามเหมือนกัน
4. ข้อสอบประกอบไปด้วย 4 ข้อใหญ่ คะแนนเต็ม 100 คะแนน โดยมีรายละเอียดดังนี้
  - 4.1 ) ข้อ 1 Hangman หน้า 1 – 4 ( 30 คะแนน )
  - 4.2 ) ข้อ 2 ห้องสมุด หน้า 4 – 9 ( 38 คะแนน )
  - 4.3) ข้อ 3 ปรับปรุง Code ร้านดอกไม้ หน้า 9 – 11 ( 12 คะแนน )
  - 4.4) ข้อ 4 ถามตอบ Google Forms 20. ข้อย่อย ( 20 คะแนน )
5. การส่งข้อสอบให้แนบ Link GitHub สำหรับข้อสอบเขียน Code และตอบคำถาม Choice ลงใน Google Forms
6. ถ้าผู้ตรวจพบวิธีการเขียน Code เหมือนกันจะใช้วิธีให้คะแนน โดยเอาคะแนนที่ได้ หารด้วย จำนวนคนที่ Code เหมือนกัน

### ข้อสอบ

#### 1. (30 คะแนน)

Hangman เป็นเกมคลาสสิก จัดเป็นเกมทายคำศัพท์โดยผู้เล่นสามารถเลือกหมวดคำศัพท์ที่จะเล่นและทราบเพียงแค่จำนวนตัวอักษรของคำศัพท์ที่เข้ามาเท่านั้น โดยแต่ละด่านให้ผู้เล่นเลือกทายตัวอักษรที่ผู้เล่นคาดว่าอยู่ในคำที่โจทย์ให้มา หากผู้เล่นทายตัวอักษรผิดหรือไม่มีในศัพท์ที่โจทย์ให้มา ผู้เล่นก็จะถูกแขวนคอ โดยเริ่มจากหัว ลำตัว แขนซ้าย แขนขวา ขาซ้าย ขาขวา ตามลำดับ เมื่อผิดครบทั้ง 6 ครั้ง จะถือว่าเป็นการแขวนคออย่างเต็มตัว ผู้เล่นก็จะแพ้ในเกมนั้น แต่ถ้าผู้เล่นทายตัวอักษรได้ถูกต้องก็จะชนะในเกมนี้ไป

จากข้อมูลดังกล่าวให้นักศึกษาสร้างเกม Hangman ขึ้นมาโดยมีรายละเอียดดังต่อไปนี้

- 1.) หน้าแรกของเกมจะแสดงเมนูโดยมีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้าแรก

Welcome to Hangman Game

- 
1. Play game
  2. Exit

Please Select Menu :

### คำอธิบายการใช้งานหน้าแรก

- 1.) เมื่อผู้ใช้งาน input menu หน้า keyboard เป็น 1 จะเข้าไปยังหน้าจอเกม Hangman
- 2.) เมื่อผู้ใช้งาน input menu หน้า keyboard เป็น 2 จะจบการทำงาน

- 2.) หน้าเกม Hangman มีรายละเอียดดังต่อไปนี้

### ตัวอย่างหน้าเกม Hangman

Play Game Hangman

-----  
Incorrect Score: 0

-----  
Input letter alphabet:

### คำอธิบายการใช้งานหน้าเกม Hangman

- 1.) เริ่มต้นเกมจะทำการสุ่มคำศัพท์ขึ้นมาจากนั้นแปลงคำศัพท์ให้เป็น \_ เพื่อให้ผู้เล่นใส่ตัวอักษรที่ผู้เล่นคิดว่ามีอยู่ในคำศัพท์ เช่น "Tennis" ตอนเริ่มต้นเกมก็จะแสดง \_ \_ \_ \_ \_
- 2.) เมื่อผู้เล่นใส่ตัวอักษรที่มีในคำศัพท์เข้าไปเช่นใส่ n ที่ Console ก็จะมีการ Clear Screen ของเก่าและทำการแสดง Console ใหม่ดังต่อไปนี้

### ตัวอย่าง Console ที่แสดงผล

Play Game Hangman

-----  
Incorrect Score: 0

\_ \_ n n \_ \_

Input letter alphabet:

- 3.) ถ้าผู้เล่นใส่ตัวอักษรผิด Incorrect Score จะเพิ่มขึ้นทีละ 1 โดยทุกครั้งที่ใส่ผิด Console จะทำการ Clear Screen และแสดงผลใหม่ดังตัวอย่าง

### ตัวอย่าง Console ที่แสดงผล

Play Game Hangman

-----

Incorrect Score: 1

\_ \_ n n \_ \_

Input letter alphabet:

- 4.) ถ้าผู้เล่นใส่ตัวอักษรผิดครบ 6 ครั้งจะขึ้น “Game Over” แล้วออกจากระบบ
- 5.) ถ้าผู้เล่นตอบครบและถูกต้อง จะขึ้นคำว่า You win!! และออกจากระบบ ดังตัวอย่าง Console ที่แสดงผล

### ตัวอย่าง Console ที่แสดงผล

Play Game Hangman

-----

Incorrect Score: 1

t e n n i s

Input letter alphabet:

Your win!!

ให้นักศึกษาเขียนโปรแกรมเพื่อแสดงผลตามขั้นตอนที่ได้อธิบายในเบื้องต้นโดยมีรายละเอียดเพิ่มเติมดังต่อไปนี้

- 1.) กำหนดคำศัพท์ที่ใช้ในการ Random ดังต่อไปนี้ “Tennis”, “Football”, “Badminton”
- 2.) ตัวอย่างการใช้งานโค้ดการ Random โดยผลลัพธ์ที่ได้จากการ Random คือตัวเลขที่อยู่ในช่วงค่าเริ่มต้นถึงค่าสิ้นสุด

```
Random random = new Random();
int resultRandom = random.Next(ค่าเริ่มต้น, ค่าสิ้นสุด);
```

- 3.) คำสั่งในการหาจำนวน ตัวอักษรใน String คือ `word.Length`
- 4.) คำสั่งในการแปลง String เป็น char array คือ `word.ToCharArray()`
- 5.) คำสั่งในการแปลง char array เป็น string คือ

```
string newGuessWord = new string(guessWordArray);
```

### เกณฑ์การให้คะแนน

- 1.) โปรแกรมสามารถรันและทำงานได้ถูกต้องตามที่โจทย์กำหนด โดยให้นักศึกษาเขียน Comment และอธิบาย Code ที่เขียนมาทุกบรรทัดด้วย (10 คะแนน)
- 2.) นักศึกษาสามารถออกแบบ Class ได้มีความเหมาะสมตามพื้นฐานการเขียนโปรแกรมเชิงวัตถุ (10 คะแนน)
- 3.) นักศึกษาสามารถเขียนโปรแกรมแสดงเมนูตามที่โจทย์กำหนดได้ถูกต้อง ( 2 คะแนน )
- 4.) นักศึกษาสามารถ Random คำศัพท์ เมื่อกด เล่นเกมได้ ( 2 คะแนน )
- 5.) นักศึกษาสามารถตรวจสอบการเล่นเกมที่ว่าผู้เล่นใส่ตัวอักษรผิดหรือไม่ ( 2 คะแนน )
- 6.) นักศึกษาสามารถ เปลี่ยนข้อความ \_ \_ \_ \_ \_ เป็นตัวอักษรที่ผู้เล่นที่ใส่เข้าไปแล้วตรงกับคำศัพท์ที่ถูกสุ่มมาได้ถูกต้อง ( 4 คะแนน )

### 2. (38 คะแนน)

ห้องสมุด มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี มีความต้องการพัฒนา Application สำหรับใช้งานภายในห้องสมุดโดยมีรายละเอียดดังต่อไปนี้

1. ระบบลงทะเบียน มีรายละเอียดดังต่อไปนี้
  - 1.1) ชื่อผู้ใช้
  - 1.2) รหัสผ่าน
  - 1.3) ประเภท สามารถใส่ 1 = Student หรือ 2 = Employee ได้
2. ระบบการเข้าสู่ระบบ มีรายละเอียดดังต่อไปนี้
  - 2.1) ชื่อผู้ใช้
  - 2.2) รหัสผ่าน
  - 2.3) ระบบจะต้องทำการตรวจสอบว่าชื่อผู้ใช้กับรหัสผ่านได้ทำการลงทะเบียนไว้หรือไม่
3. ระบบแสดงผลเมนู มีรายละเอียดดังต่อไปนี้
  - 3.1) เมนูสำหรับ นักเรียน ประกอบไปด้วย
    - ยืมหนังสือ
  - 3.2) เมนูสำหรับ พนักงานประกอบไปด้วย
    - แสดง Stock หนังสือ
4. ข้อมูลหนังสือมีรายละเอียดดังต่อไปนี้
  - 4.1) ID 1 ชื่อหนังสือ Now I understand

- 4.2) ID 2 ชื่อหนังสือ Revolutionary Wealth
- 4.3) ID 3 ชื่อหนังสือ Six Degrees
- 4.4) ID 4 ชื่อหนังสือ Les Vacances

จากข้อมูลดังกล่าวให้นักศึกษาสร้างโปรแกรมโดยมีรายละเอียดดังต่อไปนี้

- 1.) หน้าแรกของโปรแกรมจะแสดงเมนูโดยมีรายละเอียดดังนี้

#### ตัวอย่างหน้าจอแรก

Welcome to Digital Library

-----

1. Login

2. Register

Select Menu:

#### คำอธิบายการใช้งานหน้าจอแรก

- 1.) เมื่อผู้ใช้งาน input menu หน้า keyboard เป็น 2 จะเข้าไปยังหน้า Register
- 2.) เมื่อผู้ใช้งาน input menu หน้า keyboard เป็น 1 จะเข้าไปยังหน้า Login

- 2.) หน้าลงทะเบียนเพื่อเข้าใช้งานระบบมีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้าลงทะเบียน (เมื่อเลือก Type เป็นนักเรียน)

Register new Person

-----

Input name: kittinun

Input Password: 1234

Input User Type 1 = Student, 2 = Employee: 1

Student ID: 62130701103

#### ตัวอย่างหน้าลงทะเบียน (เมื่อเลือก Type เป็นพนักงาน)

Register new Person

-----

Input name: kittinun

Input Password: 1234

Input User Type 1 = Student, 2 = Employee: 2

Employee ID: 5214785

### คำอธิบายการใช้งานลงทะเบียน

- 1.) ข้อมูลที่ผู้ใช้จะต้องทำการลงทะเบียนได้แก่ ชื่อ , รหัสผ่าน, Type ของ User โดย Type ของ User จะประกอบไปด้วย ถ้าใส่ 1 มาคือ Type Student ถ้าใส่ 2 มาคือ Type Employee
- 2.) ระบบจะต้องตรวจสอบว่า ถ้า user input type เป็น 1 จะต้องใส่ข้อมูล Student ID เพิ่มเติม และถ้าใส่ type เป็น 2 จะต้องใส่ข้อมูล EmployeeID เพิ่มเติม
- 3.) เมื่อทำการลงทะเบียนเสร็จสิ้นจะเก็บข้อมูลลงใน List เพื่อใช้ในการตรวจสอบเวลาทำการ Login
- 4.) เมื่อลงทะเบียนเสร็จสิ้นจะกลับไปยังหน้าแรกที่แสดงเมนูหลัก เพื่อทำการ Login เข้าสู่ระบบต่อไป

- 3.) หน้า Login เข้าสู่ระบบมีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้า Login

Login Screen

-----

Input name: kittinun

Input Password: 1234

#### คำอธิบายการใช้งานหน้า Login

- 1.) ข้อมูลที่ผู้ใช้จะต้องทำการกรอกคือ name กับ password ที่ได้ทำการลงทะเบียนไว้
- 2.) ระบบจะต้องตรวจสอบว่า ชื่อ และ รหัสผ่านถูกต้องหรือไม่ ถ้าไม่ถูกต้องจะเปลี่ยนกลับไปหน้าแรก
- 3.) ระบบต้องตรวจสอบว่า ถ้ายังไม่ลงทะเบียนจะกลับไปยังหน้าแรก
- 4.) ระบบต้องตรวจสอบว่าถ้า login สำเร็จ คนที่เข้าสู่ระบบเป็นนักเรียนหรือพนักงาน ถ้าเป็นนักเรียนจะแสดงหน้าเมื่อนักเรียน ถ้าเป็นพนักงานจะแสดงเมนูสำหรับพนักงาน

- 4.) หน้า เมนูสำหรับ Employee เข้าสู่ระบบมีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้า เมนูสำหรับ Employee

Employee Management

-----

Name: kittinun

Employee ID: 5214785

-----

1. Get List Books

Input Menu:

### คำอธิบายการใช้งานหน้า เมนู Employee

- 1.) ข้อมูลที่ต้องแสดงคือข้อมูลของผู้ใช้ที่ทำการ Login ได้แก่ ชื่อ และ Employee ID
- 2.) แสดง เมนู เพื่อเข้าไปดูข้อมูลหนังสือทั้งหมด
- 3.) ผู้ใช้สามารถใส่เลขเมนู เพื่อไปยังหน้า รายการหนังสือได้

- 5.) หน้า รายการหนังสือ มีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้ารายการหนังสือ

Book List

-----

Book ID: 1

Book name: NOW I UNDERSTAND

Book ID: 2

Book name: REVOLUTIONARY WEALTH

Book ID: 3

Book name: Six Degrees

Book ID: 4

Book name: Les Vacances

#### คำอธิบายการใช้งานหน้ารายการหนังสือ

- 1.) แสดงรายการหนังสือทั้งหมดที่โจทย์ได้กำหนดไว้ โดยแสดง ID และชื่อหนังสือที่อยู่ในระบบทั้งหมดออกมา

- 6.) หน้า เมนูสำหรับนักเรียน มีรายละเอียดดังต่อไปนี้

#### ตัวอย่างหน้า เมนูสำหรับ Student

Student Management

-----

Name: kittinun

Student ID: 52217504

-----

1. Borrow Book

Input Menu:

### คำอธิบายการใช้งานหน้า เมนู Student

- 1.) ข้อมูลที่ต้องแสดงคือข้อมูลของผู้ใช้ที่ทำการ Login ได้แก่ ชื่อ และ Student ID
- 2.) แสดง เมนู เพื่อเข้าไปยืมหนังสือ
- 3.) ผู้ใช้สามารถใส่เลขเมนู เพื่อเข้าไปยืมหนังสือ

- 7.) หน้า การยืมหนังสือ มีรายละเอียดดังต่อไปนี้

### ตัวอย่างหน้าการยืมหนังสือ

#### Book List

```

-----
Book ID: 1
Book name: NOW I UNDERSTAND
Book ID: 2
Book name: REVOLUTIONARY WEALTH
Book ID: 3
Book name: Six Degrees
Book ID: 4
Book name: Les Vacances
Input book ID:
  
```

### คำอธิบายการใช้งานหน้า การยืมหนังสือ

- 1.) แสดงรายการยืมหนังสือที่สามารถยืมได้ทั้งหมดประกอบได้ด้วย ID และ ชื่อหนังสือ
- 2.) ผู้ใช้สามารถ Input Book ID เพื่อทำการยืมหนังสือได้ โดยเมื่อ input book id สามารถใส่ได้เรื่อยๆจนกว่าจะพิมพ์คำว่า exit จะแสดงหน้าสรุปรายละเอียดของนักเรียนและแสดงข้อมูลหนังสือที่ยืมมาทั้งหมด ดังตัวอย่างด้านล่าง

Student name: 1

Student ID: 1

Book List

-----

Book ID: 2

Book name: REVOLUTIONARY WEALTH

Book ID: 1

Book name: NOW I UNDERSTAND

Book ID: 3

Book name: Six Degrees



### เกณฑ์การให้คะแนน

- 1.) โปรแกรมสามารถรันและทำงานได้ถูกต้องตามที่โจทย์กำหนด โดยให้นักศึกษาเขียน Comment และอธิบาย Code ที่เขียนมาทุกบรรทัดด้วย (15 คะแนน)
- 2.) นักศึกษาสามารถออกแบบ Class ได้มีความเหมาะสมตามพื้นฐานการเขียนโปรแกรมเชิงวัตถุ (10 คะแนน)
- 3.) นักศึกษาสามารถเขียนโปรแกรมสำหรับลงทะเบียนสมาชิกใหม่ได้ถูกต้อง ( 2 คะแนน )
- 4.) นักศึกษาสามารถเขียนโปรแกรมเพื่อ Login เข้าสู่ระบบได้ถูกต้อง ( 2 คะแนน )
- 5.) นักศึกษาสามารถแยกเมนูแสดงผล แบบบนนักเรียน กับแบบพนักงานได้หลังทำการ Login ( 2 คะแนน )
- 6.) นักศึกษาสามารถแสดงรายการหนังสือทั้งหมดเมื่อ Login และดูรายการหนังสือที่อยู่ในระบบจาก Type Employee ได้ ( 2 คะแนน )
- 7.) นักศึกษาสามารถสร้างระบบยืมคืนหลังจาก Login ได้ตามโจทย์ที่ได้กำหนดไว้ ( 5 คะแนน)

### 3.) (12 คะแนน)

ให้นักศึกษาปรับปรุง คุณภาพ Code จากตัวอย่าง Code ที่ให้ดังต่อไปนี้

```
class Program
{
    static void Main(string[] args)
    {
        string decide = "y";
        string selectFlower;
        FlowerStore flowerStore = new FlowerStore();
        do{
            Console.WriteLine("Select number for buy flower :");
            foreach (string i in flowerStore.flowerList){
                Console.WriteLine(flowerStore.flowerList.IndexOf(i) + 1) + " ";
                Console.WriteLine(i);
            }

            selectFlower = Console.ReadLine();

            switch(selectFlower) {
```

```

        case "1":
            flowerStore.addToCart(flowerStore.flowerList[0]);
            Console.WriteLine("Added " + flowerStore.flowerList[0]);
            break;

        case "2":
            flowerStore.addToCart(flowerStore.flowerList[1]);
            Console.WriteLine("Added " + flowerStore.flowerList[1]);
            break;

        default:
            Console.WriteLine("Not Added to cart. found select number of flower");
            break;
    }

    Console.WriteLine("You can stop this progress ? exit for >> exit << progress and press
any key for continue");
    decide = Console.ReadLine();
    if(decide == "exit") {
        Console.WriteLine("Current my cart");
        flowerStore.showCart();
    }
}

while(decide != "exit");
}
}

```

```

class FlowerStore {

    public List<string> flowerList = new List<string>();
    List<string> cart = new List<string>();

    public FlowerStore(){
        flowerList.Add("Rose");
    }
}

```

```

        flowerList.Add("Lotus");
    }

    public void addToCart(string name){
        cart.Add(name);
    }

    public void showCart(){

        if(cart.Count == 0) {
            Console.WriteLine("Cart is empty");
        }else{
            Console.WriteLine("My Cart :");
            foreach (string i in cart){
                Console.WriteLine(i);
            }
        }
    }
}

```

### เกณฑ์การให้คะแนน

- 1.) มีการประยุกต์ใช้เรื่อง function เพื่อให้ Code สามารถอ่านได้ง่ายขึ้น (5 คะแนน)
- 2.) มีการนำความรู้เรื่องการเขียนโปรแกรมเชิงวัตถุมาประยุกต์ใช้งาน ( 7 คะแนน )