

ชื่อ-นามสกุล: รหัสนักศึกษา:

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

แบบทดสอบย่อยครั้งที่ 1 ภาคการศึกษาที่ 1/2567

รายวิชา: ENGCE306 Web Programming

เวลาสอบ: 180 นาที

คะแนนเต็ม: 100 คะแนน

คำชี้แจง:

1. แบบทดสอบนี้มีทั้งหมด 9 ข้อ
- 2.ให้อ่านคำถามและตอบลงในกระดาษคำตอบนี้ให้สมบูรณ์และตรงประเด็น
3. ไม่อนุญาตให้นำอุปกรณ์สื่อสารทุกชนิดเข้าห้องสอบ
4. การทุจริตในการสอบมีโทษปรับตกในรายวิชานี้

ส่วนที่ 1: ความเข้าใจพื้นฐาน (ข้อละ 5 คะแนน)

ข้อ 1. (5 คะแนน) อธิบายส่วนประกอบหลักของโครงสร้างเอกสาร HTML5 พื้นฐาน มาอย่างน้อย 3 ส่วน พร้อมบอกหน้าที่ของแต่ละส่วนโดยสังเขป

ข้อ 2. (5 คะแนน) จงบอกความแตกต่างที่สำคัญระหว่าง ID Selector และ Class Selector ใน CSS ในแง่ของการนำไปใช้งานกับองค์ประกอบ HTML

ข้อ 3. (5 คะแนน) ในภาษา JavaScript (มาตรฐาน ES6) การประกาศตัวแปรด้วยคีย์เวิร์ด let และ const มีความแตกต่างกันอย่างไรในการกำหนดและเปลี่ยนแปลงค่า?

ข้อ 4. (5 คะแนน) TypeScript คืออะไร และมีประโยชน์หลักอย่างไรเมื่อเทียบกับการใช้ JavaScript เพียงอย่างเดียวในการพัฒนาเว็บแอปพลิเคชัน?

ชื่อ-นามสกุล: รหัสนักศึกษา:

ส่วนที่ 2: การอ่านโค้ดและคิดวิเคราะห์ (ข้อละ 15-20 คะแนน)

ข้อ 5. (15 คะแนน) การใช้ Semantic HTML (เช่น <article>, <section>, <nav>) มีข้อดีกว่าการใช้แท็ก <div> เพียงอย่างเดียวในการจัดโครงสร้างเว็บไซต์อย่างไร? จงอธิบายในมุมมองของ SEO (Search Engine Optimization) และ Accessibility (การเข้าถึงสำหรับผู้พิการ)

ข้อ 6. (15 คะแนน) จากโค้ด HTML และ CSS ที่กำหนดให้ข้างล่างนี้ จงอธิบายว่าแท็ก <p> ที่มีข้อความว่า "This paragraph is special." จะถูกแสดงผลบนหน้าจออย่างไรบ้าง? (ให้อธิบายคุณสมบัติด้านสไตล์ เช่น สี, ความหนา, พื้นหลัง ที่ได้รับผลมาจาก CSS)

```
<!DOCTYPE html>
<html lang="th">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ตัวอย่างข้อสอบข้อ 6</title>
  <style>
    #main-content {
      font-family: Arial, sans-serif;
    }
    .highlight {
      background-color: yellow;
    }
    p.special {
      color: red;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div id="main-content">
    <h1>Article Title</h1>
    <p class="highlight">This is a standard paragraph.</p>
    <p class="highlight special">This paragraph is special.</p>
  </div>
</body>
</html>
```

ชื่อ-นามสกุล: รหัสนักศึกษา:

ข้อ 7. (20 คะแนน) จากโค้ด JavaScript ต่อไปนี้ จงอธิบายการทำงานที่ละขั้นตอนเมื่อฟังก์ชัน `setupCounter` ถูกเรียกใช้ และอธิบายว่า "Closure" มีบทบาทสำคัญอย่างไรที่ทำให้ตัวแปร `count` ยังคงค่าเดิมไว้ได้ทุกครั้งที่ถูกใช้คลิกปุ่ม

```
function setupCounter() {  
  let count = 0;  
  const counterButton = document.getElementById('myButton');  
  
  counterButton.addEventListener('click', function() {  
    count++;  
    counterButton.textContent = `Clicked ${count} times`;  
  });  
}  
  
setupCounter();
```

ชื่อ-นามสกุล: รหัสนักศึกษา:

ข้อ 8. (20 คะแนน) พิจารณาโค้ด TypeScript ต่อไปนี้ และตอบคำถาม:

1. อธิบายความสัมพันธ์ระหว่าง interface Movable, class Vehicle, และ class Car ว่าแต่ละส่วนเกี่ยวข้องกับอย่างไรผ่านคีย์เวิร์ด implements และ extends
2. หากมีการสร้าง instance `const myCar = new Car('Toyota', 4);` แล้วเรียกใช้ `myCar.move();` ผลลัพธ์ที่แสดงใน Console คืออะไร และเพราะเหตุใด?

```
interface Movable {
  speed: number;
  move(): void;
}

class Vehicle {
  constructor(public brand: string) {}

  move(): void {
    console.log("A vehicle is moving.");
  }
}

class Car extends Vehicle implements Movable {
  speed: number = 60;

  constructor(brand: string, public doors: number) {
    super(brand);
  }

  // Overriding the move method
  move(): void {
    console.log(`The ${this.brand} car is moving at ${this.speed} km/h.`);
  }
}
```

ชื่อ-นามสกุล: รหัสนักศึกษา:

ข้อ 9. (15 คะแนน) จากโค้ด JavaScript ด้านล่าง จงอธิบายการทำงานที่ละขั้นตอนว่าทำไมผลลัพธ์สุดท้ายของ activeUserNames จึงเป็น ['ALICE', 'CHARLIE']? ให้อธิบายการทำงานของ for...of loop ร่วมกับเงื่อนไข if และการทำงานของเมธอด. map()

```
const userData = [
  { name: 'Alice', age: 25, isActive: true },
  { name: 'Bob', age: 17, isActive: false },
  { name: 'Charlie', age: 32, isActive: true },
  { name: 'David', age: 15, isActive: true }
];

const activeUsers = [];
for (const user of userData) {
  if (user.age >= 18 && user.isActive) {
    activeUsers.push(user);
  }
}

const activeUserNames = activeUsers.map(user => user.name.toUpperCase());

console.log(activeUserNames);
```