

# Week 4 Quiz Solutions [11 points]

(Taken on Week 6)

## 1) Promise (3 points)

Given this API call:

```
fetch('https://api.example.com/data')
```

Write JS code to handle:

- A successful response [1.5 pt]
- An error [1.5 pt]

```
// Handling successful response (1.5pt)
fetch('https://api.example.com/data')
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json();
  })
  .then(data => {
    console.log('Success:', data);
  });

// Error handling (1.5pt)
  .catch(error => {
    console.error('Error:', error);
  });
```

### Explanation:

- First `.then()` checks response status and parses JSON
- Second `.then()` handles the successful data
- `.catch()` handles any errors in the chain

## 2) Hashing (2 points)

About SHA-256 hashing:

- Explain why we hash passwords [1 pt]
- What's the length of a SHA-256 hash? [1 pt]

### Why hash passwords: [1pt]

Hashing passwords converts them into irreversible, fixed-length strings. This protects user credentials by:

- Preventing plaintext storage (security breach protection)
- Making it computationally infeasible to reverse-engineer
- Allowing verification without storing actual passwords

### SHA-256 hash length: [1pt]

64 characters (hexadecimal representation of 256 bits)

### 3) Jasmine (2 points)

Given this function:

```
function giveMeTwo() { return 2 }
```

Write a Jasmine test to verify it returns 2 [2 pt]

```
describe('giveMeTwo function', () => {  
  it('should return 2', () => {  
    const result = giveMeTwo();  
    expect(result).toBe(2);  
  });  
});
```

#### Explanation:

- describe() groups related tests
- it() defines a single test case
- expect().toBe() verifies exact value match
- Full points awarded for correct syntax and assertion

### 4) Jasmine Testing (4 points)

Given this function:

```
function isAdult(age) { return age >= 18; }
```

Write Jasmine test cases:

- Test for true when age  $\geq 18$  [1.5 pt]
- Test for false when age  $< 18$  [1.5 pt]
- Correct Jasmine syntax [1 pt]

```
describe('isAdult function', () => {  
  // Test for true case (1.5pt)  
  it('should return true for adults (18+)', () => {  
    expect(isAdult(18)).toBeTrue();  
    expect(isAdult(25)).toBeTrue();  
  });  
  
  // Test for false case (1.5pt)  
  it('should return false for minors (<18)', () => {  
    expect(isAdult(17)).toBeFalse();  
    expect(isAdult(10)).toBeFalse();  
  });  
});
```

#### Explanation:

- Two test cases covering both scenarios (1.5pt each)
- Uses toBeTrue() and toBeFalse() matchers
- Proper Jasmine structure (1pt for syntax)
- Includes multiple examples within each test case