Started on	Thursday, 7 September 2023, 11:08 PM
State	Finished
Completed on	Thursday, 7 September 2023, 11:29 PM
Time taken	21 mins 27 secs
Marks	9.00/12.00
Grade	**7.50** out of 10.00 (**75**%)

QUESTION 1

Correct

Mark 1.00 out of 1.00

What is the output of the following code:

```
function job() {
    return new Promise(function(resolve, reject) {
        reject();
    });
}

let promise = job();

promise
.then(function() {
    console.log('Success 1');
})
.then(function() {
    console.log('Success 2');
})
.then(function() {
    console.log('Success 3');
})
.catch(function() {
    console.log('Error 1');
})
.then(function() {
    console.log('Success 4');
});
```

Select one:

- a. Success 1, Success 2, Success 3, Error 1, Success 4
- b. Error 1, Success 4 ✔
- c. Success 1, Success 2, Success 3, Success 4
- d. Error 1, Success 1, Success 2, Success 3, Success 4
- e. Error 1
- f. Success 1, Error 1

QUESTION 2

Correct

Mark 1.00 out of 1.00

What is the output of the following code:

```
function job(state) {
    return new Promise(function(resolve, reject) {
        if (state) {
            resolve('success');
        } else {
            reject('error');
        }
    });
}

let promise = job(true);

promise
.then(function(data) {
    console.log(data);
    return job(false);
})
.catch(function(error) {
    console.log(error);
    return 'Error caught';
})
.then(function(data) {
    console.log(data);
    return job(true);
})
.catch(function(error) {
    console.log(error);
});
```

Select one:

- a. error, Error caught, success
- b. success, success
- c. error, success, Error caught
- d. success, error, Error caught ✔
- e. success, error, success, error

QUESTION 3

Correct

Mark 1.00 out of 1.00

What are 2 native functions to run code asynchronously in JavaScript ?

Select one or more:

- [x] a. setTimeout ✔
- [ ] b. interval
- [ ] c. repeat
- [x] d. setInterval ✔
- [ ] e. startInterval
- [ ] f. timeout
- [ ] g. delay

QUESTION 4

Correct

Mark 1.00 out of 1.00

What is the output of the following code?

```
let fs = require('fs');

console.log('1');

fs.readFile('test.txt', 'utf8', function(error, data) {
    if (error) {
        throw error;
    }
    console.log('2');
});

console.log('3');
```

Select one:

- ( ) a. 2 1 3
- ( ) b. 1 2 3
- ( ) c. 3 2 1
- ( ) d. 2 3 1
- (•) e. 1 3 2 ✔

QUESTION 5

Correct

Mark 1.00 out of 1.00

What is the function to stop an interval timer?

Select one:

- a. clearInterval ✔
- b. clearTimer
- c. shutdownTimer
- d. stopTimer

QUESTION 6

Incorrect

Mark 0.00 out of 1.00

What will be displayed to the console when calling the `f()` function?

```
async function f() {
  let result = 'first!';
  let promise = new Promise((resolve, reject) => {
    setTimeout(() => resolve('done!'), 1000);
  });

  result = await promise;

  console.log(result);
}

f();
```

Select one:

- a. Something else ✖
- b. JavaScript error
- c. first!
- d. done!

QUESTION 7

Correct

Mark 1.00 out of 1.00

What is the output of the following code?

```
function* generator(i) {
  yield i;
  yield i * 2;
}

const gen = generator(10);

console.log(gen.next().value);
console.log(gen.next().value);
```

Select one:

- a. 10, 20 ✔
- b. [0, 10], [10, 20]
- c. 20, 20
- d. 0, 10 and 10, 20

QUESTION 8

Incorrect

Mark 0.00 out of 1.00

What is the output of the following code?

```
const myPromise = () => Promise.resolve('I have resolved!');

function firstFunction() {
  myPromise().then(res => console.log(res));
  console.log('second');
}

async function secondFunction() {
  console.log(await myPromise());
  console.log('second');
}

firstFunction();
secondFunction();
```

Select one:

- a. I have resolved!, second and second, I have resolved!
- b. second, I have resolved! and second, I have resolved!
- c. I have resolved!, second and I have resolved!, second ✖
- d. second, I have resolved! and I have resolved!, second

QUESTION 9

Correct

Mark 1.00 out of 1.00

What's the output of the following code?

```
async function* range(start, end) {
  for (let i = start; i ≤ end; i++) {
    yield Promise.resolve(i);
  }
}

(async () ⇒ {
  const gen = range(1, 3);
  for await (const item of gen) {
    console.log(item);
  }
})();
```

Select one:

- a. 1 2 3 ✔
- b. undefined undefined undefined
- c. Promise {1} Promise {2} Promise {3}
- d. Promise {<pending>} Promise {<pending>} Promise {<pending>}

QUESTION 10

Correct

Mark 1.00 out of 1.00

Select the missing line of code

```
const teams = [
  { name: 'Team 1', members: ['Paul', 'Lisa'] },
  { name: 'Team 2', members: ['Laura', 'Tim'] },
];

function* getMembers(members) {
  for (let i = 0; i < members.length; i++) {
    yield members[i];
  }
}

function* getTeams(teams) {
  for (let i = 0; i < teams.length; i++) {
    // ✨ SOMETHING IS MISSING HERE ✨
  }
}

const obj = getTeams(teams);
obj.next(); // { value: "Paul", done: false }
obj.next(); // { value: "Lisa", done: false }
```

Select one:

- a. yield getMembers(teams[i].members)
- b. return yield getMembers(teams[i].members)
- c. return getMembers(teams[i].members)
- d. yield* getMembers(teams[i].members) ✔

QUESTION 11

Incorrect

Mark 0.00 out of 1.00

What types of errors can we handle with the try..catch construct?

Select one:

- a. logical errors
- b. runtime errors
- c. all of the above ✖
- d. critical errors
- e. syntax errors

QUESTION 12

Correct

Mark 1.00 out of 1.00

What will be the result of executing?

(*sum* was not defined neither before nor after the given code.)

```
try {
    console.log("Start program");
    let result  =  sum(10, 20);
    console.log("End program");
} catch(error) {
    console.log("We caught a bug!");
} finally{
    console.log("Finish!");
}
```

Select one:

a.
Start program
We caught a bug!

b.
Start program
End program
Finish!

c.
Start program
End program
We caught a bug!
Finish!

d.
Start program
End program
We caught a bug!

e.
Start program
Finish!

f. ✔
Start program
We caught a bug!
Finish!

◄ Tasks. Asynchronous JS, Closures, Exceptions

Jump to...

Slides. Scrum Fundamental ►