Started on	Saturday, 30 September 2023, 9:39 AM
State	Finished
Completed on	Saturday, 30 September 2023, 10:00 AM
Time taken	21 mins 7 secs
Marks	11.50/13.00
Grade	**2.65** out of 3.00 (**88.46**%)

QUESTION **1**

Correct

Mark 1.00 out of 1.00

Please, fill in the gaps, so the code will compile:

```
interface Lengthwise {
  length: number;
}

function loggingIdentity< Type ✓ extends ✓ Lengthwise ✓ >(arg: Type): Type {
  console.log(arg.length);
  return arg;
}
```

uses |

QUESTION **2**

Correct

Mark 1.00 out of 1.00

Suppose, we have variable, defined like this:

```
const course: Course = {
  name: "Math",
  totalHours: 50;
  id: 0,
};
```

You need to define type Course, please, select missing words:

```
interface ✓ Course {
  name: string ✓ ;
  totalHours: number ✓
  id: number ✓ ;
}
```

Please, use the most specific suitable data types.

alias define

never string any number

QUESTION 3

Correct

Mark 1.00 out of 1.00

Please, create a type that reflects the task status with one of three possible values: "todo", "implementing" and "done".

type ✓ TaskStatus ="todo" ✓ | ✓ "implementing" ✓ | ✓ "done"; ✓

class interface

"implementing" string = any ;

& |

QUESTION 4

Correct

Mark 1.00 out of 1.00

To start a new Create React App project with TypeScript, we can run:

npx create-react-app *app-name* --template typescript ✓

QUESTION 5

Correct

Mark 1.00 out of 1.00

Is the code below correct? What is the result of compilation of this code?

```
function loggingIdentity<Type>(arg: Type): Type {
  console.log(arg.length);
  return arg;
}
```

● Compilation error because property 'length' does not exist on type 'Type'. ✓

○ No compilation errors

○ Compilation error due to incorrect definition of function

QUESTION 6

Correct

Mark 1.00 out of 1.00

Please, select how we can annotate a function that takes no arguments and returns a value of type Course

○ function getMainCourse() {
//...
} => Course

● function getMainCourse(): Course { ✓
//...
}

○ function getMainCourse() {
//...
}: Course

○ function getMainCourse() => Course {
//...
}

QUESTION 7

Correct

Mark 1.00 out of 1.00

Please, create a type definition for an array where each value is an object with *name* property of *string* type

○ type ObjectsWithNamesArray = Array<object>;

○ type ObjectsWithNamesArray = Array<name:string>;

● type ObjectsWithNamesArray = Array<{ name: string }>; ✓

○ type ObjectsWithNamesArray = name:Array<string>;

QUESTION 8

Partially correct

Mark 0.50 out of 1.00

Please, select the type the following statement is about:

A type representing a concept of React Element - representation of a native DOM component (e.g. `<div />`), or a user-defined composite component (e.g. `<MyComponent />`)

☑ React.ReactElement ✓

□ JSX.Element

□ React.ReactNode

□ React.Component

□ React.FC

QUESTION 9

Correct

Mark 1.00 out of 1.00

Please, select the type representing any possible type of React node (ReactElement + primitive JS types)

○ React.FC

○ JSX.Element

○ React.Component

● React.ReactNode ✓

QUESTION 10

Correct

Mark 1.00 out of 1.00

Will the next code compile? Please, choose the correct option.

```
interface Person {
  name: string;
  surname: string;
}

function logPerson(p: Person) {
  console.log(`${p.name}, ${p.surname}`);
}

const person = { name: "Paul", surname: "Mass" };
logPerson(person);
```

○ The code will not compile because we must specify the type of the argument passed to a function when invoke the function.

● The code will compile because the structure argument passed to a function corresponds to Person interface ✓

○ The code will not compile because the argument passed to a function is not defined as variable of type Person

QUESTION 11

Correct

Mark 1.00 out of 1.00

Please, choose how the TypeScript works with the code like this:

```
interface Course {
  name: string;
  totalHours: number;
  id: number;
}
```

```
const course: Course = {
  name: "Math",
  totalHours: 50
};
```

● Compilation error because variable is not assignable to type 'Course' ✓

○ Compilation error because type Course is defined incorrectly

○ No compilation errors

QUESTION 12

Correct

Mark 1.00 out of 1.00

Suppose, you have a variable defined like this:

```
let helloWorld = "Hello World";
```

What statement is correct?

○ The type of the variable will be inferred as *any*

○ We can not define variables like this with TypeScript, we must provide a type name.

○ The type of the variable will be inferred as *never*

● The type of the variable will be inferred as *string* ✓

QUESTION 13

Incorrect

Mark 0.00 out of 1.00

Will the next code compile with TypeScript?

```
  let s = "hello"
  s = 1;
```

● Yes, we can reassign value of another type to a variable, because we didn't specify the type in declaration ✗

○ No, we will get an error that type 'number' is not assignable to type 'string'

○ No, we will get an error that variable has not specified type at declaration

◄ Quiz. Tools: npm, Webpack, Babel, ESLInt

Jump to...

Practical tasks. Tools: npm, Webpack, Babel, ESLInt, Next.js, Typescript ►