AppDev2

Section Id :	64065339716
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065384393
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 140 Question Id : 640653587048 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MODERN APPLICATION DEVELOPMENT II (COMPUTER BASED EXAM) "

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531958834. ✓ YES

6406531958835. ✗ NO

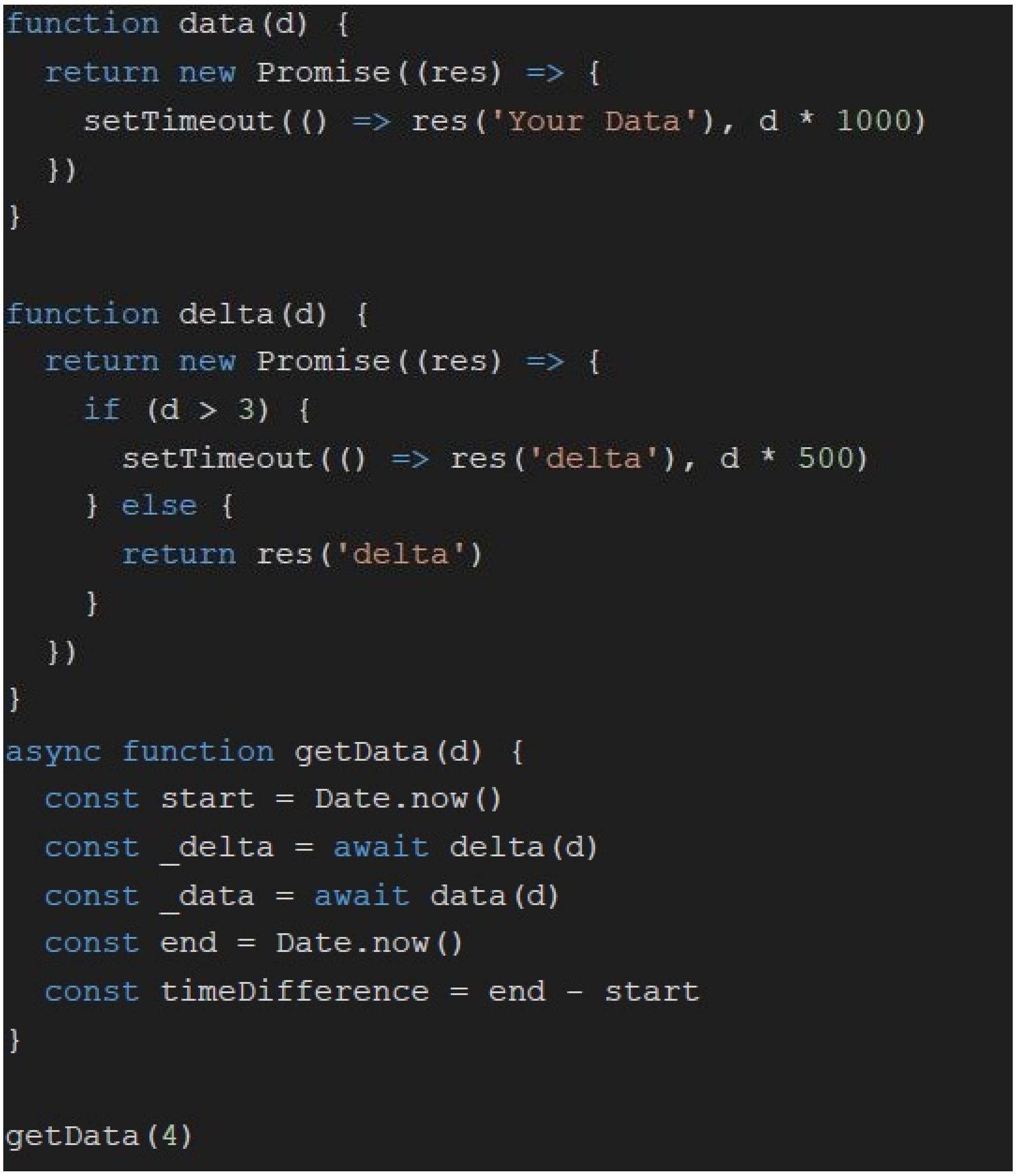
Sub-Section Number :	2
Sub-Section Id :	64065384394
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 141 Question Id : 640653587049 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the below javascript program.



```
function data(d) {
  return new Promise((res) => {
    setTimeout(() => res('Your Data'), d * 1000)
  })
}

function delta(d) {
  return new Promise((res) => {
    if (d > 3) {
      setTimeout(() => res('delta'), d * 500)
    } else {
      return res('delta')
    }
  })
}
async function getData(d) {
  const start = Date.now()
  const _delta = await delta(d)
  const _data = await data(d)
  const end = Date.now()
  const timeDifference = end - start
}

getData(4)
```

What will be the approximate value of the variable “timeDifference”?

Options :

6406531958836. ✖ Less than 4000

6406531958837. ✖ Is equal to 4000

6406531958838. ✔ Greater than or equal to 6000

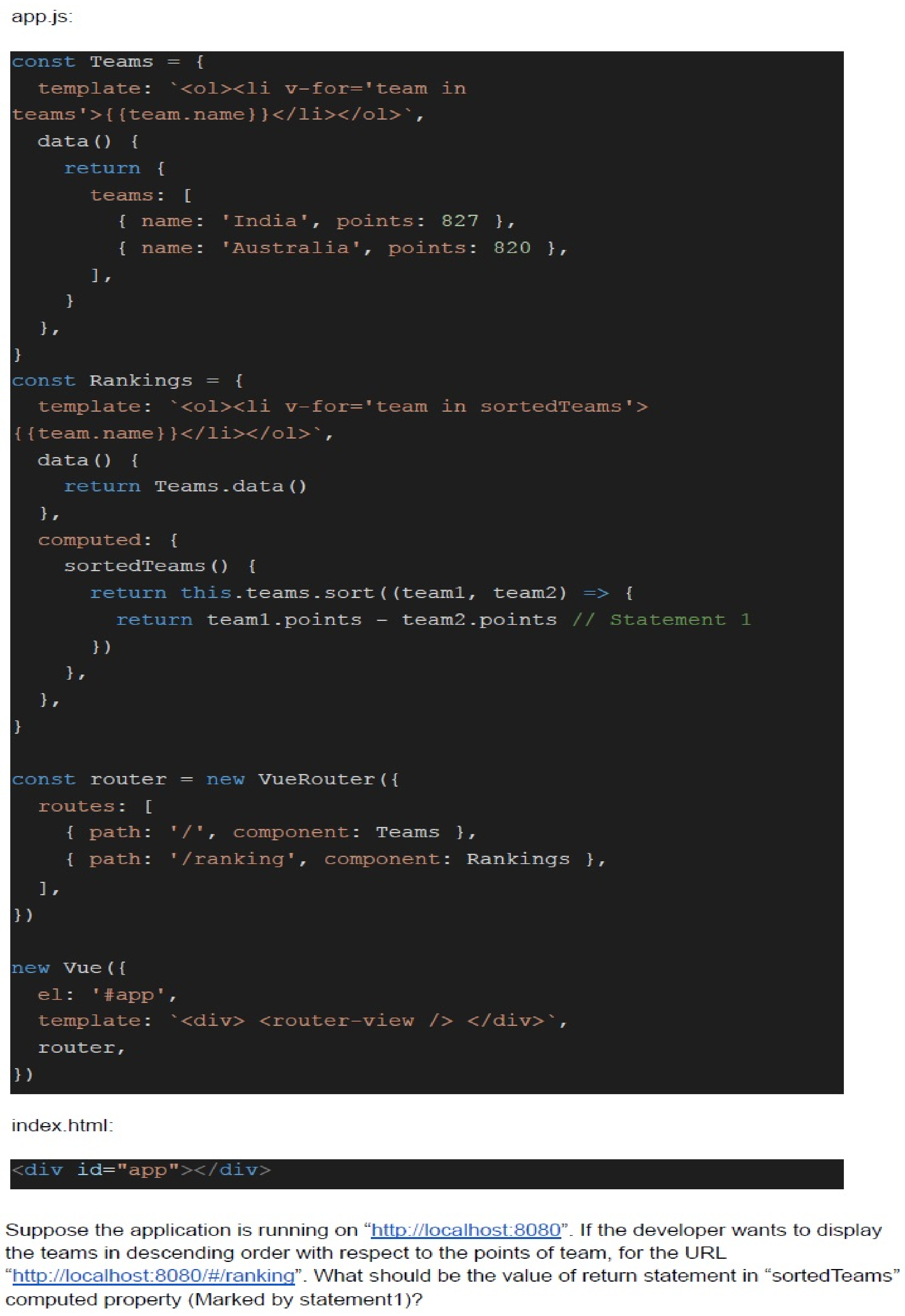
6406531958839. ✖ None of these

Question Number : 142 Question Id : 640653587052 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Vue application with javascript file “app.js” and markup file “index.html”.



app.js:

```
const Teams = {
  template: `<ol><li v-for='team in
teams'>{{team.name}}</li></ol>`,
  data() {
    return {
      teams: [
        { name: 'India', points: 827 },
        { name: 'Australia', points: 820 },
      ],
    }
  },
}
const Rankings = {
  template: `<ol><li v-for='team in sortedTeams'>
{{team.name}}</li></ol>`,
  data() {
    return Teams.data()
  },
  computed: {
    sortedTeams() {
      return this.teams.sort((team1, team2) => {
        return team1.points - team2.points // Statement 1
      })
    },
  },
}

const router = new VueRouter({
  routes: [
    { path: '/', component: Teams },
    { path: '/ranking', component: Rankings },
  ],
})

new Vue({
  el: '#app',
  template: `<div> <router-view /> </div>`,
  router,
})
```

index.html:

```
<div id="app"></div>
```

Suppose the application is running on “http://localhost:8080”. If the developer wants to display the teams in descending order with respect to the points of team, for the URL “http://localhost:8080/#/ranking”. What should be the value of return statement in “sortedTeams” computed property (Marked by statement1)?

Options :

6406531958848. ✖ team1.points - team2.points

6406531958849. ✓ team2.points - team1.points

6406531958850. ✗ team1 > team2

6406531958851. ✗ team2 < team1

Question Number : 143 Question Id : 640653587053 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

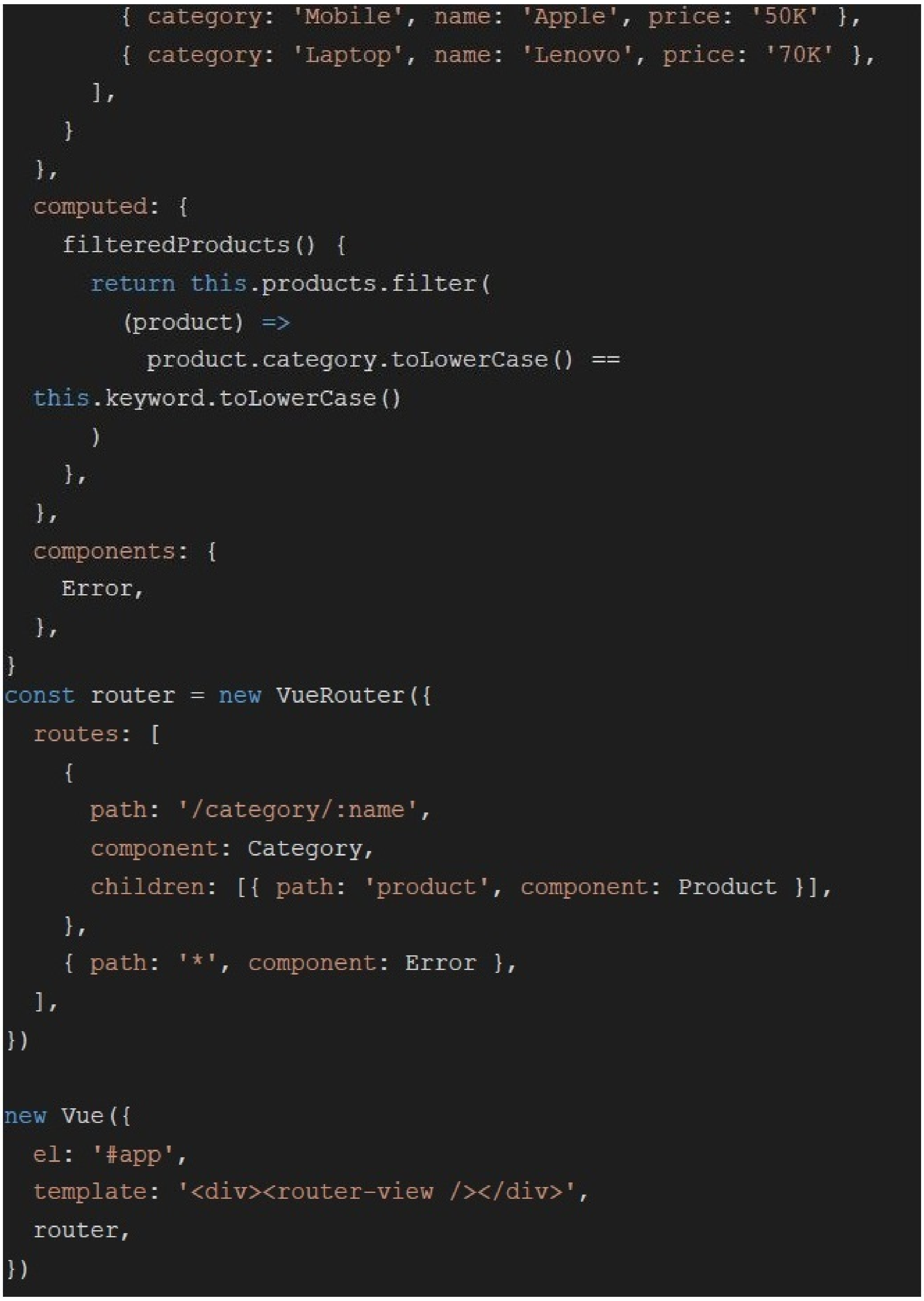
Consider the following Vue application with javascript file “app.js” and markup file “index.html”.

index.html:

```html
<body>
    <div id="app"></div>
</body>
```

app.js:

```javascript
const Category = {
  template:
  `<div>{{$route.params.name}}<router-view></router-view></div>
  `,
}

const Error = {
  template: `<div><slot> Page Not Found </slot></div>`,
}
const Product = {
  template: `
  <div>
    <div v-if='filteredProducts.length > 0'>
        <div v-for='product in filteredProducts'>
            Name: {{product.name}}, Price: {{product.price}}
        </div>
    </div>
    <Error v-else> No {{keyword}} Found </Error>
  </div>`,
  data() {
    return {
      keyword: this.$route.params.name,
      products: [
        { category: 'Mobile', name: 'Samsung', price: '10K' },
```



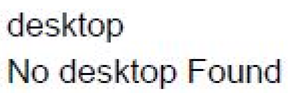
```
        { category: 'Mobile', name: 'Apple', price: '50K' },
        { category: 'Laptop', name: 'Lenovo', price: '70K' },
      ],
    }
  },
  computed: {
    filteredProducts() {
      return this.products.filter(
        (product) =>
          product.category.toLowerCase() ==
  this.keyword.toLowerCase()
      )
    },
  },
  components: {
    Error,
  },
}
const router = new VueRouter({
  routes: [
    {
      path: '/category/:name',
      component: Category,
      children: [{ path: 'product', component: Product }],
    },
    { path: '*', component: Error },
  ],
})

new Vue({
  el: '#app',
  template: '<div><router-view /></div>',
  router,
})
```

Suppose the application is running on “http://localhost:8080”. What will be rendered by the browser for the URL “http://localhost:8080/#/”?

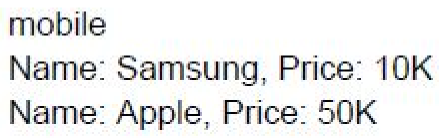
Options :

6406531958852. ✓ Page Not Found

6406531958853. ✗ laptop
Name: Lenovo, Price: 70K

6406531958854. ✖ 

```
desktop
No desktop Found
```

6406531958855. ✖ 

```
mobile
Name: Samsung, Price: 10K
Name: Apple, Price: 50K
```

Question Number : 144 Question Id : 640653587055 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

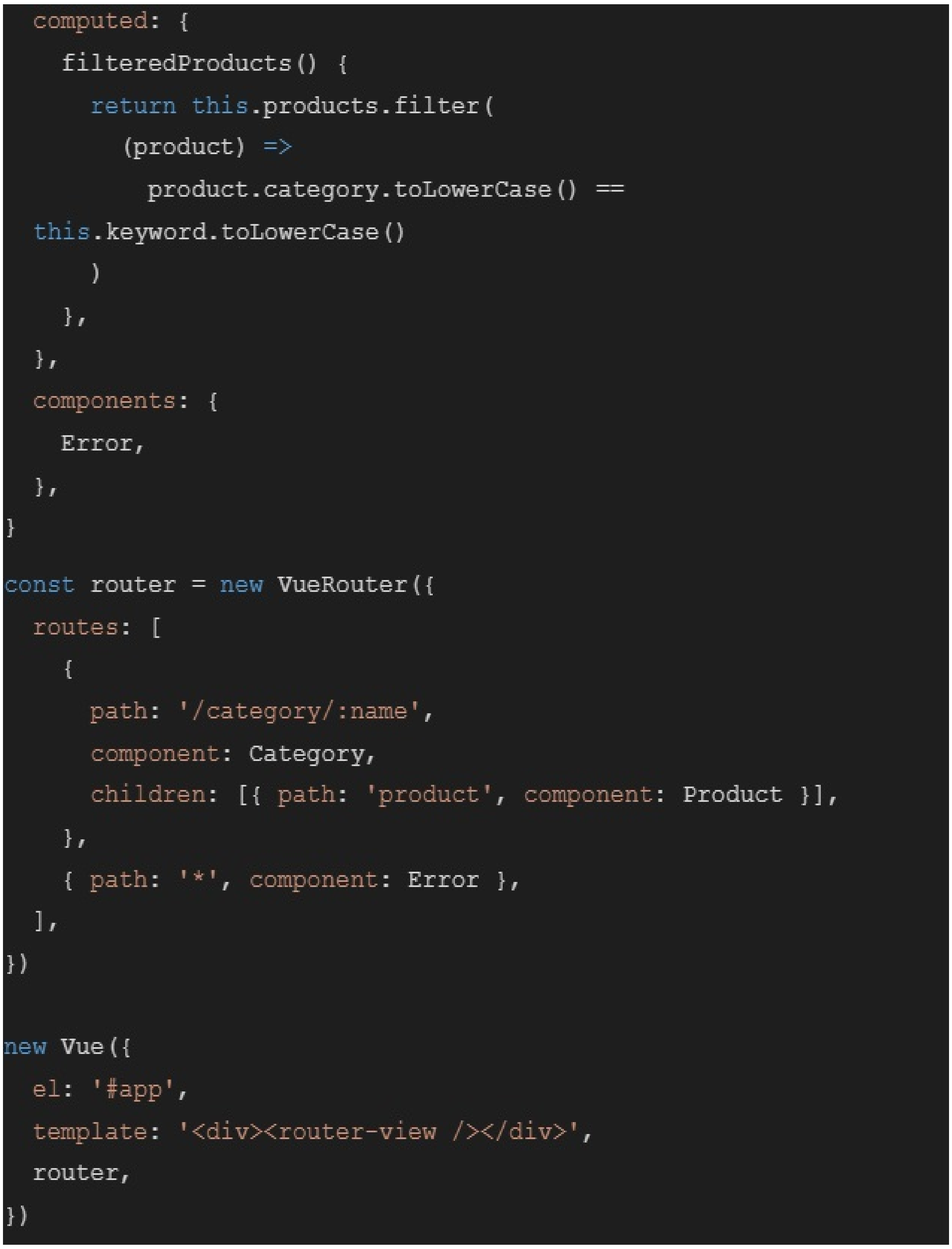
Consider the following Vue application with javascript file “app.js” and markup file “index.html”.

index.html:

```html
<body>
    <div id="app"></div>
</body>
```

app.js:

```javascript
const Category = {
  template:
  `<div>{{$route.params.name}}<router-view></router-view></div>`,
}

const Error = {
  template: `<div><slot> Page Not Found </slot></div>`,
}

const Product = {
  template: `
  <div>
    <div v-if='filteredProducts.length > 0'>
        <div v-for='product in filteredProducts'>
            Name: {{product.name}}, Price: {{product.price}}
        </div>
    </div>
    <Error v-else> No {{keyword}} Found </Error>
  </div>`,
  data() {
    return {
      keyword: this.$route.params.name,
      products: [
        { category: 'Mobile', name: 'Samsung', price: '10K' },
        { category: 'Mobile', name: 'Apple', price: '50K' },
        { category: 'Laptop', name: 'Lenovo', price: '70K' },
      ],
    }
  },
```



```
  computed: {
    filteredProducts() {
      return this.products.filter(
        (product) =>
          product.category.toLowerCase() ==
  this.keyword.toLowerCase()
      )
    },
  },
  components: {
    Error,
  },
}

const router = new VueRouter({
  routes: [
    {
      path: '/category/:name',
      component: Category,
      children: [{ path: 'product', component: Product }],
    },
    { path: '*', component: Error },
  ],
})

new Vue({
  el: '#app',
  template: '<div><router-view /></div>',
  router,
})
```

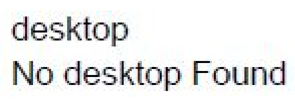
Suppose the application is running on "http://localhost:8080". What will be rendered by the browser for the URL "http://localhost:8080/#/category/desktop/product"?

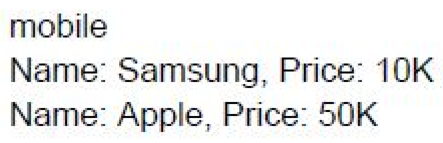
Options :

6406531958860. ✖ Page Not Found

6406531958861. ✖ laptop
Name: Lenovo, Price: 70K

6406531958862. ✔


```
desktop
No desktop Found
```

6406531958863. ✖ 
```
mobile
Name: Samsung, Price: 10K
Name: Apple, Price: 50K
```

Question Number : 145 Question Id : 640653587056 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Vue application with javascript file “app.js” and markup file “index.html”.

index.html

```
<body>
    <div id="app"></div>
</body>
```

app.js

```
const Product = {
  template: `<div>
    <slot name="title">No title</slot>
    <slot>No Description Given</slot>
    <slot name="price">Flexible</slot>
    </div>`,
}
new Vue({
  el: '#app',
  template: `
  <Product>
    <p>
       <b>Lenovo Desktop</b>
       8GB RAM, 512GB SSD
    </p>
    <template v-slot:price>
       61K
    </template>
  </Product>`,
  components: {
    Product,
  },
})
```

Suppose the application is running on “http://localhost:8080”. What will be rendered by the browser for the URL “http://localhost:8080/#/”?

Options :

6406531958864. ✓

No title
Lenovo Desktop 8GB RAM, 512GB SSD
61K

6406531958865. ✗

Lenovo Desktop
8GB RAM, 512GB SSD
61K

6406531958866. ✖ **Lenovo Desktop**
8GB RAM, 512GB SSD
Flexible

6406531958867. ✖ No title
Lenovo Desktop 8GB RAM, 512GB SSD
Flexible

Question Number : 146 Question Id : 640653587060 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the below javascript program, and predict the output, if executed.

```
const data = {
    count: 10
};

const newData = {}

Object.defineProperty(newData, 'count', {
    get() {
        return data.count;
    },
    set(newValue) {
        data.count = newValue;
    },
});
console.log("Before Update:", newData.count);
newData.count = 20;
console.log("After Update:", data.count);
```

Options :

6406531958880. ✗ Before Update: 10

After Update: 10

6406531958881. ✗ Before Update: 10

Error

6406531958882. ✓ Before Update: 10

After Update: 20

6406531958883. ✗ Before Update: 10

After Update: undefined

Sub-Section Number :	3
Sub-Section Id :	64065384395
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 147 Question Id : 640653587050 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following is correct regarding the finally block of Promise?

Options :

6406531958840. ✗ It allows scheduling a function after the promise is rejected.

6406531958841. ✗ It allows scheduling a function after the promise is resolved.

6406531958842. ✓ It allows scheduling a function when promise is either fulfilled or rejected.

6406531958843. ✗ None of these

Question Number : 148 Question Id : 640653587063 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following statements is true?

Options :

6406531958892. ✗ The data stored in session storage gets lost as soon as the page is refreshed.

6406531958893. ✗ The data stored in local storage gets lost as soon as the system shuts down.

6406531958894. ✓ The data stored in session cookies gets lost as soon as the browser is closed.

6406531958895. ✗ All of these

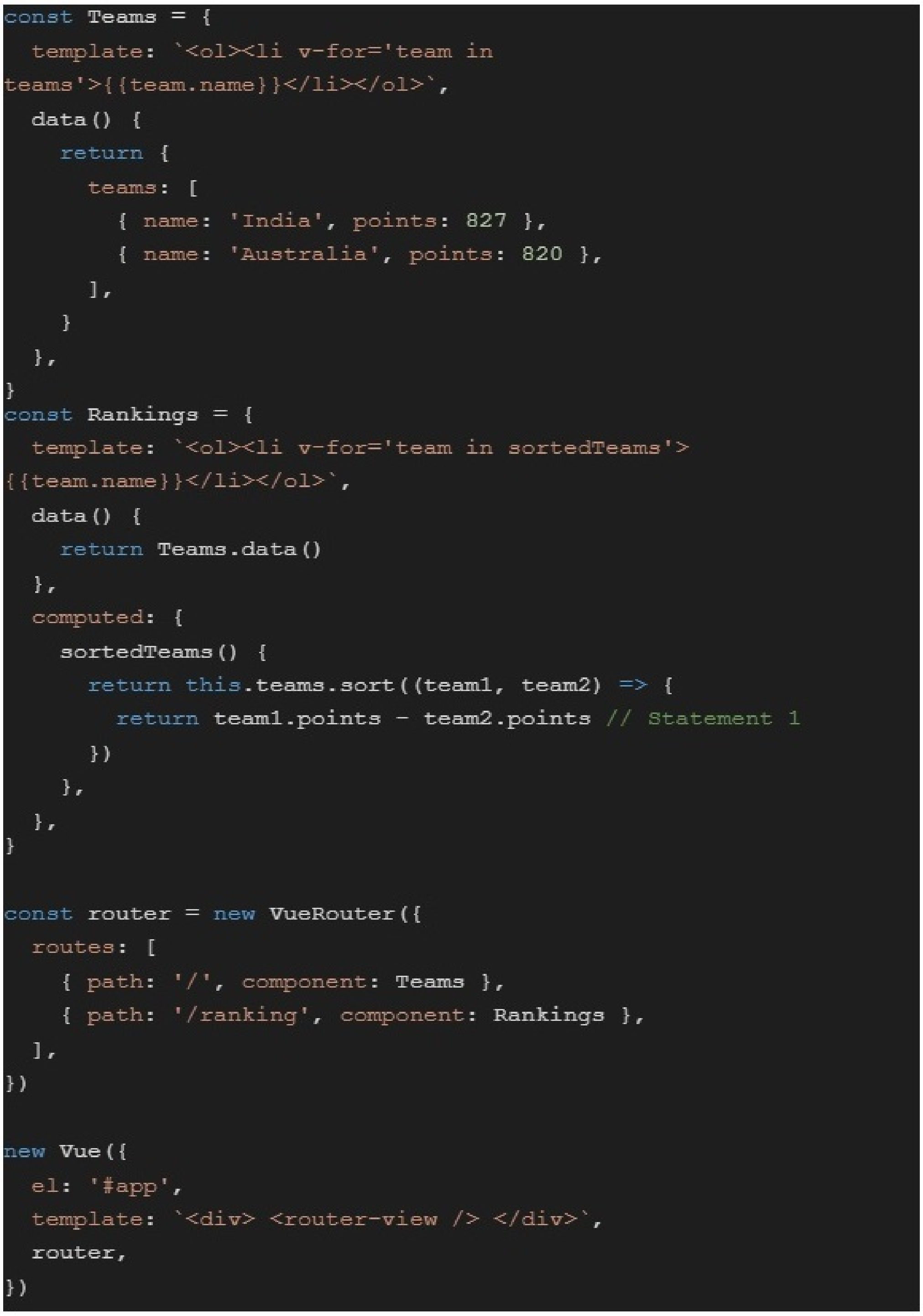
Sub-Section Number :	4
Sub-Section Id :	64065384396
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 149 Question Id : 640653587051 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

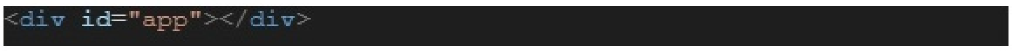
Question Label : Multiple Choice Question

Consider the following Vue application with javascript file “app.js” and markup file “index.html”.



```
const Teams = {
  template: `<ol><li v-for='team in
teams'>{{team.name}}</li></ol>`,
  data() {
    return {
      teams: [
        { name: 'India', points: 827 },
        { name: 'Australia', points: 820 },
      ],
    }
  },
}
const Rankings = {
  template: `<ol><li v-for='team in sortedTeams'>
{{team.name}}</li></ol>`,
  data() {
    return Teams.data()
  },
  computed: {
    sortedTeams() {
      return this.teams.sort((team1, team2) => {
        return team1.points - team2.points // Statement 1
      })
    },
  },
}

const router = new VueRouter({
  routes: [
    { path: '/', component: Teams },
    { path: '/ranking', component: Rankings },
  ],
})

new Vue({
  el: '#app',
  template: `<div> <router-view /> </div>`,
  router,
})
```

index.html:



```
<div id="app"></div>
```

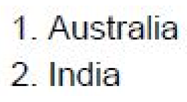
Suppose the application is running on “http://localhost:8080”. What will be rendered by the browser for the url “http://localhost:8080/#/ranking”

Options :

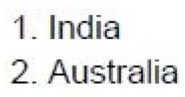
6406531958844. ✖ Australia

6406531958845. ✖ India

6406531958846.

✓ 

1. Australia
2. India

6406531958847. ✖ 

1. India
2. Australia

Question Number : 150 Question Id : 640653587054 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question

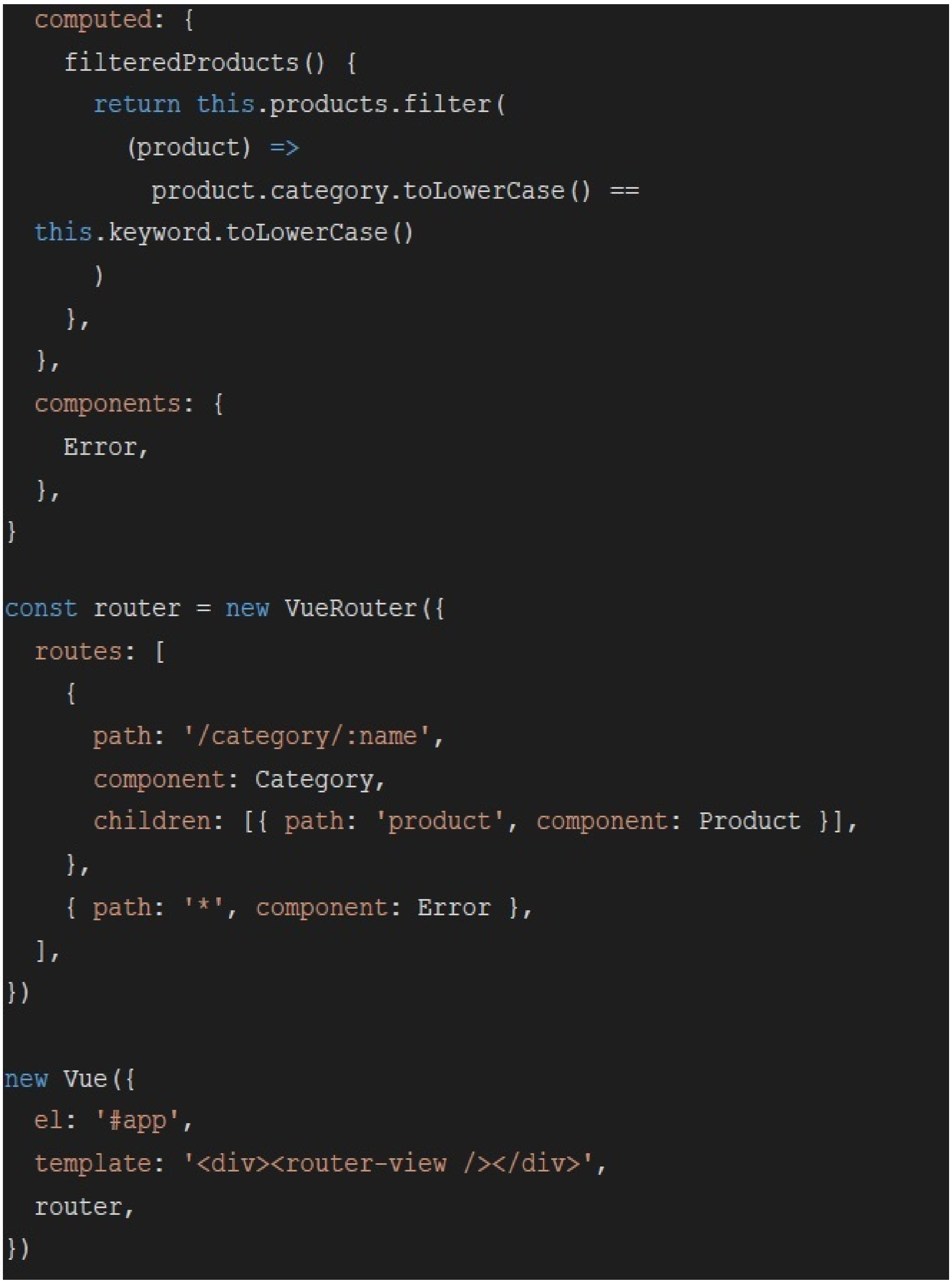
Consider the following Vue application with javascript file “app.js” and markup file “index.html”.

index.html:

```
<body>
    <div id="app"></div>
</body>
```

app.js:

```
const Category = {
  template:
  `<div>{{$route.params.name}}<router-view></router-view></div>
  `,
}

const Error = {
  template: `<div><slot> Page Not Found </slot></div>`,
}


const Product = {
  template: `
  <div>
    <div v-if='filteredProducts.length > 0'>
        <div v-for='product in filteredProducts'>
            Name: {{product.name}}, Price: {{product.price}}
        </div>
    </div>
    <Error v-else> No {{keyword}} Found </Error>
  </div>`,
  data() {
    return {
      keyword: this.$route.params.name,
      products: [
        { category: 'Mobile', name: 'Samsung', price: '10K' },
        { category: 'Mobile', name: 'Apple', price: '50K' },
        { category: 'Laptop', name: 'Lenovo', price: '70K' },
      ],
    }
  },
```



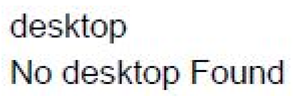
```
  computed: {
    filteredProducts() {
      return this.products.filter(
        (product) =>
          product.category.toLowerCase() ==
  this.keyword.toLowerCase()
      )
    },
  },
  components: {
    Error,
  },
}

const router = new VueRouter({
  routes: [
    {
      path: '/category/:name',
      component: Category,
      children: [{ path: 'product', component: Product }],
    },
    { path: '*', component: Error },
  ],
})

new Vue({
  el: '#app',
  template: '<div><router-view /></div>',
  router,
})
```

Suppose the application is running on “http://localhost:8080”. What will be rendered by the browser for the URL “http://localhost:8080/#/category/mobile/product”?

Options :

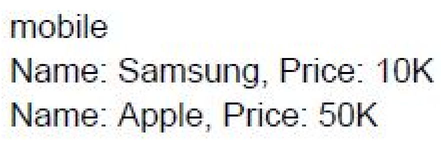
6406531958856. ✖ Page Not Found

6406531958857. ✖ laptop
Name: Lenovo, Price: 70K



6406531958858. ✖

```
desktop
No desktop Found
```



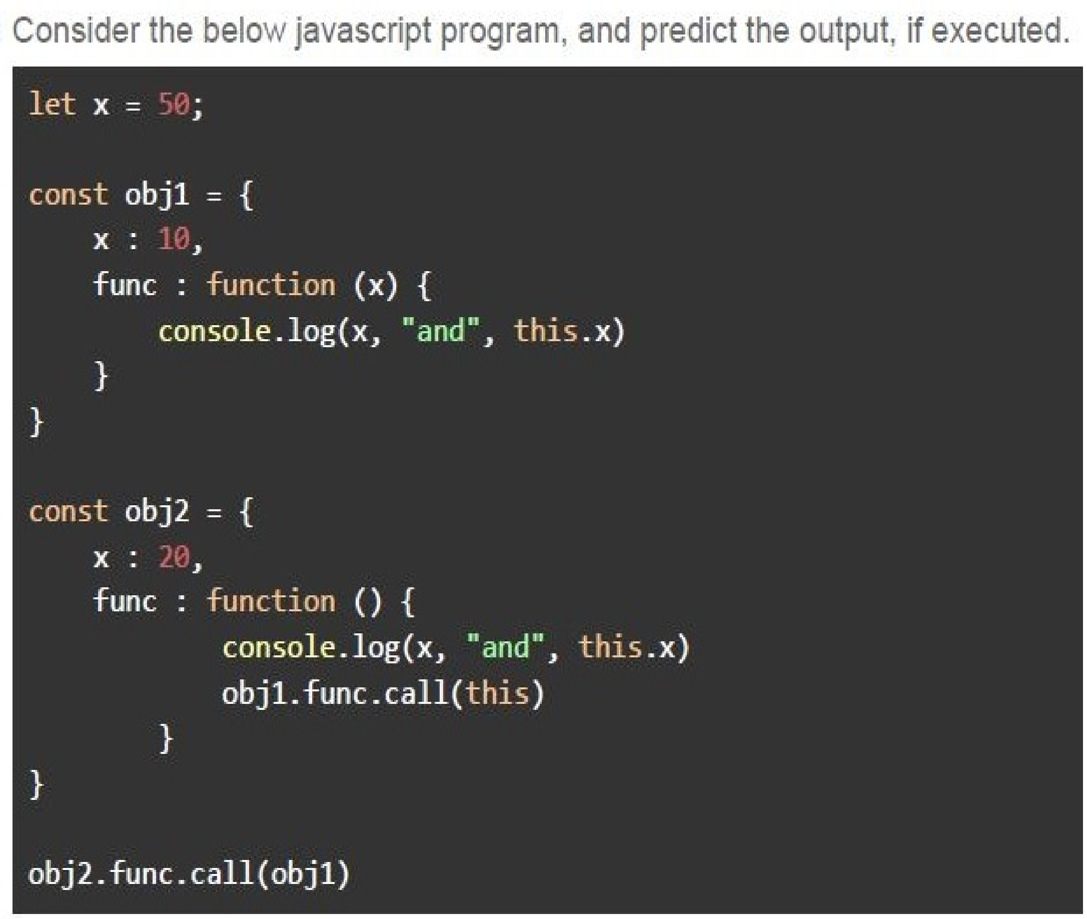
6406531958859. ✓

```
mobile
Name: Samsung, Price: 10K
Name: Apple, Price: 50K
```

Question Number : 151 Question Id : 640653587057 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question



Consider the below javascript program, and predict the output, if executed.

```
let x = 50;

const obj1 = {
    x : 10,
    func : function (x) {
        console.log(x, "and", this.x)
    }
}

const obj2 = {
    x : 20,
    func : function () {
            console.log(x, "and", this.x)
            obj1.func.call(this)
        }
}

obj2.func.call(obj1)
```

Options :

6406531958868. ✖ 50 and 10
undefined and 20

6406531958869. ✓ 50 and 10

undefined and 10

6406531958870. ✖ 50 and 20

50 and 20

6406531958871. ✖ 20 and 10

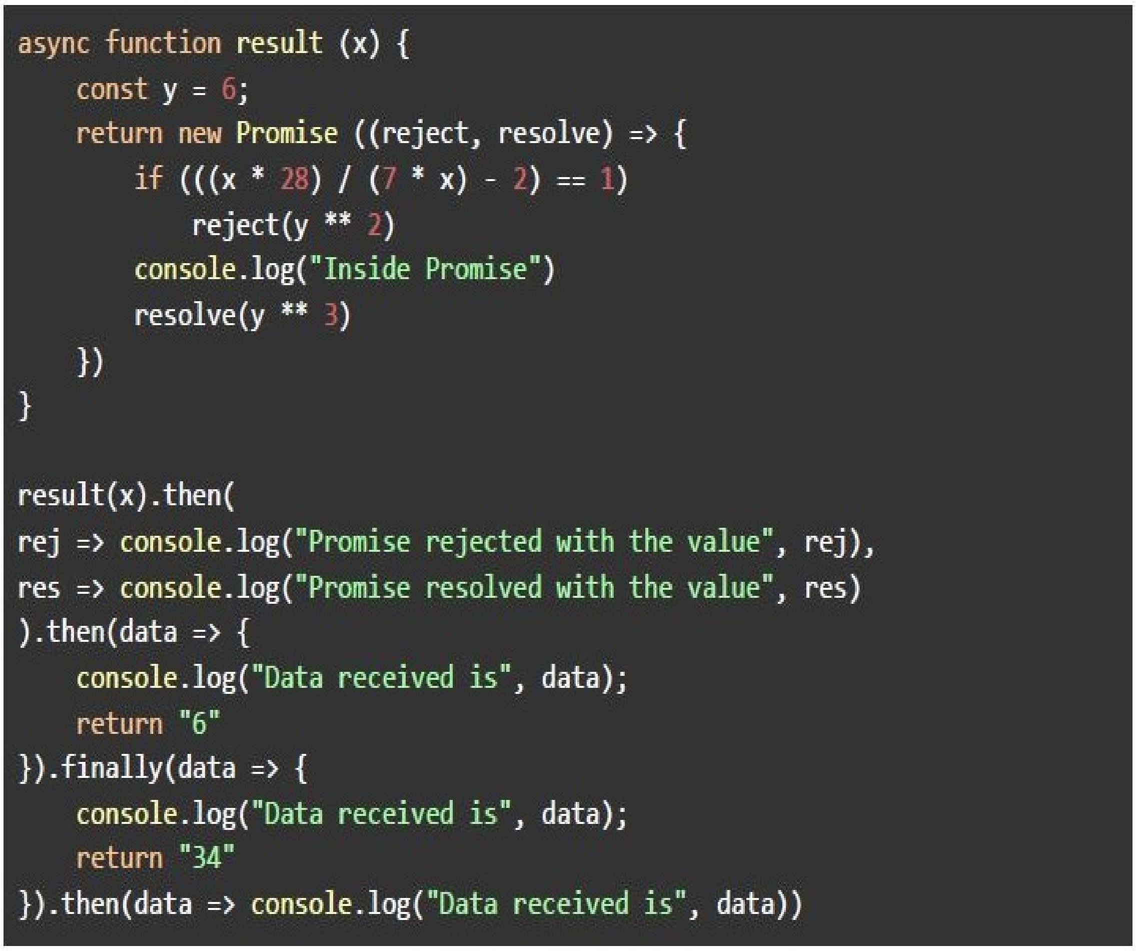
50 and 10

Question Number : 152 Question Id : 640653587062 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the below javascript program.



```
async function result (x) {
    const y = 6;
    return new Promise ((reject, resolve) => {
        if (((x * 28) / (7 * x) - 2) == 1)
            reject(y ** 2)
        console.log("Inside Promise")
        resolve(y ** 3)
    })
}

result(x).then(
rej => console.log("Promise rejected with the value", rej),
res => console.log("Promise resolved with the value", res)
).then(data => {
    console.log("Data received is", data);
    return "6"
}).finally(data => {
    console.log("Data received is", data);
    return "34"
}).then(data => console.log("Data received is", data))
```

Assuming the variable “x” passed to the “result” function is a whole number between 1 and 99 (including 1 and 99), what will be shown on the console, if the above program is executed?

Options :

6406531958888. ✖ Promise resolved with the value 36

Data received is 36

Data received is 6

Data received is 34

6406531958889. ✖ Inside Promise

Promise rejected with the value 36

Data received is undefined

Data received is undefined

Data received is 34

6406531958890. ✖ Promise rejected with the value 216

Data received is undefined

Data received is undefined

Data received is 34

6406531958891. ✓ Inside Promise

Promise rejected with the value 216

Data received is undefined

Data received is undefined

Data received is 6

Sub-Section Number :	5
Sub-Section Id :	64065384397
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 153 Question Id : 640653587058 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are true regarding javascript?

Options :

6406531958872. ✖ The “this” reference always refers to the global object inside an arrow function.

6406531958873. ✓ The “call” and “apply” functions are the same, if the calling function doesn’t accept any arguments.

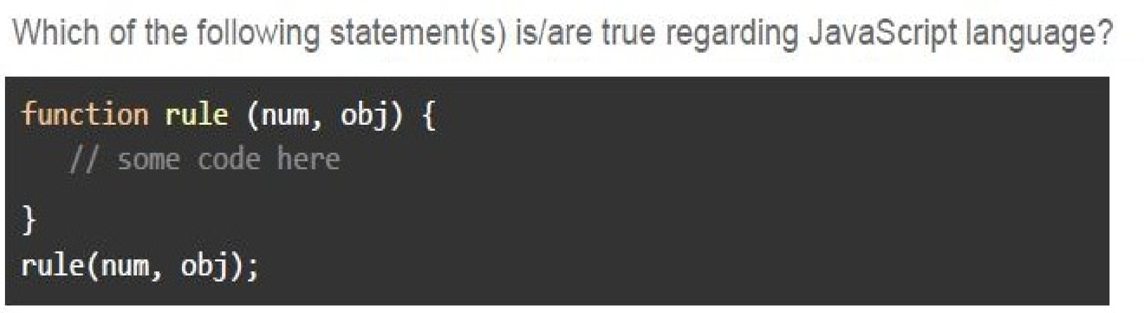
6406531958874. ✓ A variable, declared using “var” outside any function, is accessible throughout the program.

6406531958875. ✖ All of these

Question Number : 154 Question Id : 640653587061 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question



Which of the following statement(s) is/are true regarding JavaScript language?

```
function rule (num, obj) {
   // some code here
}
rule(num, obj);
```

Options :

6406531958884. ✓ In the above function call, the variable “num” is passed by value, assuming it to be holding a string literal.

6406531958885. ✓ In the above function call, the variable “obj” is passed by reference, assuming it to be holding an object.

6406531958886. ✖ A variable declared using the keyword “var” will have the value “null”, until it is initialized.

6406531958887. ✖ All of these

Sub-Section Number :	6
Sub-Section Id :	64065384398
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 155 Question Id : 640653587059 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the below Vue class bindings:

Code 1:

```
<div :class="[isActive ? 'activeClass' : '', 'errorClass']"></div>
```

Code 2:

```
<div :class="[{ 'activeClass': isActive }, 'errorClass']"></div>
```

Which of the following statement(s) is/are true?

Options :

6406531958876. ✔ Both the code snippets will render the same HTML.

6406531958877. ✖ Both the code snippets will render the different HTML.

6406531958878. ✖ The code snippet 2 will always render the element div with class “activeClass”, and “errorClass” will only be applied, if the data variable “isActive” is truthy.

6406531958879. ✔ The code snippet 1 will always render the element div with class “errorClass”, and “activeClass” will only be applied”, if the data variable “isActive” is truthy.

Question Number : 156 Question Id : 640653587064 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are correct in the context of REST and GraphQL?

Options :

6406531958896. ✖ Both REST and GraphQL are software architectural styles.

6406531958897. ✓ Both REST and GraphQL are in general used for fetching data from a remote storage.

6406531958898. ✗ Every API must adhere to all the REST principles.

6406531958899. ✓ In GraphQL, a write operation should be performed explicitly via a mutation, in general.

MLT

Section Id :	64065339717
Section Number :	11
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	13
Number of Questions to be attempted :	13
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065384399
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 157 Question Id : 640653587065 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0