

Change Background Color :	No
Change Theme :	No
Help Button :	No
Show Reports :	No
Show Progress Bar :	No

Group I

Group Number :	1
Group Id :	64065313875
Group Maximum Duration :	0
Group Minimum Duration :	90
Show Attended Group? :	No
Edit Attended Group? :	No
Break time :	0
Group Marks :	632
Is this Group for Examiner? :	No
Examiner permission :	Cant View
Show Progress Bar? :	No
Revisit allowed for group Instructions? :	Yes
Maximum Instruction Time :	0
Minimum Instruction Time :	0
Group Time In :	Minutes
Navigate To Group Summary From Last Question? :	No
Disable Submit Button During Assessment? :	No
Section Selection Time? :	0
No of Optional sections to be attempted :	0

Section Id :	64065339794
Section Number :	1
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 :	64065384922
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 1 Question Id : 640653588501 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 :

6406531962991. ✓ YES

6406531962992. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	64065384923
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 2 Question Id : 640653588502 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 :

6406531962993. ✖ Less than 4000

6406531962994. ✖ Is equal to 4000

6406531962995. ✔ Greater than or equal to 6000

6406531962996. ✖ None of these

Question Number : 3 Question Id : 640653588505 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 :

6406531963005. ✖ team1.points - team2.points

6406531963006. ✓ team2.points - team1.points

6406531963007. ✘ team1 > team2

6406531963008. ✘ team2 < team1

Question Number : 4 Question Id : 640653588506 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 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 :

6406531963009. ✓ Page Not Found

laptop

6406531963010. ✘ Name: Lenovo, Price: 70K

desktop

6406531963011. ✖ No desktop Found

mobile

Name: Samsung, Price: 10K

6406531963012. ✖ Name: Apple, Price: 50K

Question Number : 5 Question Id : 640653588508 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 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 :

6406531963017. ✖ Page Not Found

6406531963018. ✖ Name: Lenovo, Price: 70K

6406531963019. ✔

desktop

No desktop Found

mobile

Name: Samsung, Price: 10K

6406531963020. ✖ Name: Apple, Price: 50K

Question Number : 6 Question Id : 640653588509 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 :

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

6406531963021. ✓

6406531963022. ✖

Lenovo Desktop
8GB RAM, 512GB SSD
61K

6406531963023. ✖

Lenovo Desktop
8GB RAM, 512GB SSD
Flexible

6406531963024. ✖

No title
Lenovo Desktop 8GB RAM, 512GB SSD
Flexible

Question Number : 7 Question Id : 640653588513 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 :

6406531963037. ✖ Before Update: 10

After Update: 10

6406531963038. ✖ Before Update: 10

Error

6406531963039. ✔ Before Update: 10

After Update: 20

6406531963040. ✖ Before Update: 10

After Update: undefined

Sub-Section Number :	3
Sub-Section Id :	64065384924
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 8 Question Id : 640653588503 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 :

6406531962997. ✖ It allows scheduling a function after the promise is rejected.

6406531962998. ✖ It allows scheduling a function after the promise is resolved.

6406531962999. ✔ It allows scheduling a function when promise is either fulfilled or rejected.

6406531963000. ✖ None of these

Question Number : 9 Question Id : 640653588516 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 :

6406531963049. ✖ The data stored in session storage gets lost as soon as the page is refreshed.

6406531963050. ✖ The data stored in local storage gets lost as soon as the system shuts down.

6406531963051. ✔ The data stored in session cookies gets lost as soon as the browser is closed.

6406531963052. ✖ All of these

Sub-Section Number : 4

Sub-Section Id : 64065384925

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 10 Question Id : 640653588504 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 :

6406531963001. ✖ Australia

6406531963002. ✖ India

6406531963003.

- 1. Australia
- ✓ 2. India

- 1. India
 - 2. Australia
6406531963004. ✖

Question Number : 11 Question Id : 640653588507 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 :

6406531963013. ✖ Page Not Found

6406531963014. ✖ laptop
Name: Lenovo, Price: 70K

desktop

6406531963015. ✖ No desktop Found

mobile

Name: Samsung, Price: 10K

Name: Apple, Price: 50K

6406531963016. ✔

Question Number : 12 Question Id : 640653588510 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 :

6406531963025. ✖ 50 and 10

undefined and 20

6406531963026. ✔ 50 and 10

undefined and 10

6406531963027. ✖ 50 and 20

50 and 20

6406531963028. ✖ 20 and 10

50 and 10

Question Number : 13 Question Id : 640653588515 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 :

6406531963045. ✖ Promise resolved with the value 36

Data received is 36

Data received is 6

Data received is 34

6406531963046. ✖ Inside Promise

Promise rejected with the value 36

Data received is undefined

Data received is undefined

Data received is 34

6406531963047. ✖ Promise rejected with the value 216

Data received is undefined

Data received is undefined

Data received is 34

6406531963048. ✔ 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 :	64065384926
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 14 Question Id : 640653588511 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 :

6406531963029. ✖ The "this" reference always refers to the global object inside an arrow function.

6406531963030. ✓ The “call” and “apply” functions are the same, if the calling function doesn't accept any arguments.

6406531963031. ✓ A variable, declared using “var” outside any function, is accessible throughout the program.

6406531963032. ✗ All of these

Question Number : 15 Question Id : 640653588514 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 :

6406531963041. ✓ In the above function call, the variable “num” is passed by value, assuming it to be holding a string literal.

6406531963042. ✓ In the above function call, the variable “obj” is passed by reference, assuming it to be holding an object.

6406531963043. ✗ A variable declared using the keyword “var” will have the value “null”, until it is initialized.

6406531963044. ✗ All of these

Sub-Section Number :	6
Sub-Section Id :	64065384927
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 16 Question Id : 640653588512 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 :

6406531963033. ✓ Both the code snippets will render the same HTML.

6406531963034. ✖ Both the code snippets will render the different HTML.

6406531963035. ✖ 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.

6406531963036. ✓ 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 : 17 Question Id : 640653588517 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 :

6406531963053. ✖ Both REST and GraphQL are software architectural styles.

6406531963054. ✓ Both REST and GraphQL are in general used for fetching data from a remote storage.

6406531963055. ✖ Every API must adhere to all the REST principles.

6406531963056. ✓ In GraphQL, a write operation should be performed explicitly via a mutation, in general.

Sem1 English1

Section Id :	64065339795
Section Number :	2
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	23
Number of Questions to be attempted :	23
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 :	64065384928
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 18 Question Id : 640653588518 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0