

Options :

- This program generates the output:
No jobs
100 fresher jobs

6406531736948. ✖
- This program generates the output:
100 fresher jobs
No jobs

6406531736949. ✔
- This program generates the output:
No jobs
175 fresher jobs

6406531736950. ✖
- The program terminates due to unhandled exception(s).

6406531736951. ✖

AppDev2

Section Id :	64065333937
Section Number :	9
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 :	64065373955

Question Shuffling Allowed :No

Is Section Default? :null

Question Number : 132 Question Id : 640653521122 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 2"

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 :
6406531736960. ✓ YES
6406531736961. ✖ NO

Sub-Section Number :2

Sub-Section Id :64065373956

Question Shuffling Allowed :Yes

Is Section Default? :null

Question Number : 133 Question Id : 640653521126 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

Fill in code1 & code2, which can be used in Vuex Store to update the "below_average" state variable with the objects of those students who have scored less than 55 marks.

```
const store= new Vuex.Store({
  state:{
    student_total:0,
    students:[
      {
        name : 'Akshay',
        marks : 52
      },
      {
        name : 'Vishwajeet',
        marks : 78
      },
      {
        name : 'Sonali',
        marks : 43
      }
    ],
    below_average:[]
  },
  code1:{
    belowAverageStudents(state){
      code2
    },
  }
})
```

Options :

code1: mutations

```
code2: students.forEach(student=>{
  if(student.marks < 55)
    below_average.push(student)
})
```

6406531736974. ✖

6406531736975. ✖

code1: actions

```
code2: state.students.forEach(student=>{  
    if(student.mark < 55)  
        below_average.toppers.push(student)  
    })
```

code1: actions

```
code2: context.students.forEach(student=>{  
    if(student.marks < 55)  
        context.below_average.push(student)  
    })
```

6406531736976. ✖

code1: mutations

```
code2: state.students.forEach(student=>{  
    if(student.marks < 55)  
        state.below_average.push(student)  
    })
```

6406531736977. ✔

Question Number : 134 Question Id : 640653521128 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 false regarding PWAs and SPAs in the context of web?

Options :

6406531736982. ✖ A PWA is a web application that can be installed to a device's home screen.

6406531736983. ✔ An SPA is a web application that allows a user to interact with the app as usual even when offline.

6406531736984. ✖ A PWA must have service workers installed on the client.

6406531736985. ✖ YouTube is an example of a PWA.

Question Number : 135 Question Id : 640653521133 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

Consider the below javascript code.

```
async function customFetch(url) {
  try {
    console.log(url)
    const res = await fetch(url)
    if (!res.ok) {
      throw new Error(`HTTP Error: ${res.status}`)
    }
    try {
      const data = await res.json()
      console.log(data)
    } catch {
      throw new Error('Data is not JSON serializable')
    }
  } catch {
    throw new Error('Network Error')
  }
}
customFetch('https://example.com/api/users/23').catch((err) => {
  console.error(err)
})
```

Suppose the API URL "<https://example.com/api/users/23>" throws a 404 error.
What will be logged on to the console, except the URL?

Options :

- 6406531737002. ✖ HTTP Error: 404
- 6406531737003. ✖ Data is not JSON serializable
- 6406531737004. ✔ Network Error
- 6406531737005. ✖ None of these

Sub-Section Number : 3

Sub-Section Id : 64065373957

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 136 Question Id : 640653521125 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 markup "index.html" and javascript file "app.js".

index.html:

```
<div id = "app">
  <my-comp>
    <template v-slot:first = "slotProps">
      This is from {{slotProps.user.name1}} template
    </template>

    <template v-slot:default = "slotProps">
      This is from {{slotProps.user.name3}} template
    </template>

    <template v-slot:second = "slotProps">
      This is from {{slotProps.user.name2}} template
    </template>
  </my-comp>
</div>
<script src = "app.js"> </script>
```

app.js:

```
Vue.component("myComp", {
  template : `<div>
    <p>
      <slot v-bind:user="user">
      </slot>
    </p>

    <p>
      <slot name = "first" v-bind:user="user">
      </slot>
    </p>
  </div>
  `,
  data : function () {
    return {
      user : {
        'name1' : "Abhi's",
        'name2' : "Dev's",
        'name3' : "Sonali's",
      }
    }
  }
})

const app = new Vue({
  el : "#app",
})
```

Options :

6406531736970. ✖ This is from template

This is from template

6406531736971. ✖ This is from Abhi's template

This is from Dev's template

This is from Sonali's template

6406531736972. ✖ This is from Abhi's template

This is from Sonali's template

6406531736973. ✔ This is from Sonali's template

This is from Abhi's template

Question Number : 137 Question Id : 640653521130 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 javascript program, and predict the output, if executed.

```
var x = 20;

const y = {
  x : 40,
  func : function(x) {
    console.log("Value is:", x)
  }
}

const z = {
  x : 40,
  func : () => {
    console.log("Value is:", x)
    y.func.call(this)
  }
}

z.func.apply(y)
```

Options :

6406531736990. ✖ Value is: 40

Value is: undefined

6406531736991. ✖ Value is: 40

Value is: 20

6406531736992. ✖ Value is: 20

Value is: 40

6406531736993. ✔ Value is: 20

Value is: undefined

Question Number : 138 Question Id : 640653521131 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 code, and predict the output, if executed?

```
const runScored = (score) => {  
  score > 100 ? true : false  
}  
  
const teamSelector = (run) => {  
  return new Promise((res, rej) => {  
    if (runScored(run)) {  
      res('In the team')  
    } else {  
      rej('Out of the team')  
    }  
  })  
}  
  
teamSelector(100)  
  .then((res) => {  
    console.log(res)  
  })  
  .catch((res) => {  
    console.log(res)  
  })
```

Options :

6406531736994. ✖ In the team

6406531736995. ✓ Out of the team

6406531736996. ✖ Will throw an error

6406531736997. ✖ None of these

Question Number : 139 Question Id : 640653521132 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 code, and predict the output, if executed?

```
const proGen = (t) => {  
  return new Promise((res, rej) => {  
    setTimeout(() => {  
      if (t > 5) {  
        rej('Disqualified')  
      } else {  
        res('Qualified')  
      }  
    }, t * 1000)  
  })  
}  
  
async function decider() {  
  const res1 = await proGen(3)  
  console.log(res1)  
  const res2 = await proGen(6)  
  console.log(res2)  
  console.log('Decision Over')  
}  
decider().catch((data) => {  
  console.log(data)  
})  
console.log('Decision Started')
```

Options :

6406531736998. ✖ Decision Started

Qualified

Disqualified

Decision Over

6406531736999. ✖ Qualified

Disqualified

Decision Over

6406531737000. ✔ Decision Started

Qualified

Disqualified

6406531737001. ✖ None of these

Question Number : 140 Question Id : 640653521134 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 code.

```
async function customFetch(url) {
  try {
    console.log(url)
    const res = await fetch(url)
    if (!res.ok) {
      throw new Error(`HTTP Error: ${res.status}`)
    }
    try {
      const data = await res.json()
      console.log(data)
    } catch {
      throw new Error('Data is not JSON serializable')
    }
  } catch {
    throw new Error('Network Error')
  }
}
customFetch('https://example.com/api/users/23').catch((err) => {
  console.error(err)
})
```

Suppose the API URL "<https://example.com/api/users/23>" returns a valid HTML output.

What will be logged on to console, except the URL?

Options :

- 6406531737006. ✖ HTTP Error: 404
- 6406531737007. ✖ Data is not JSON serializable
- 6406531737008. ✔ Network Error
- 6406531737009. ✖ None of these

Question Number : 141 Question Id : 640653521135 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 app with markup "index.html" and javascript file "app.js".

index.html:

```
<body>
  <div id="app">
    <router-view></router-view>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3/dist/vue-router.js"></script>
  <script src="app.js"></script>
</body>
```

app.js:

```
const Home = { template: `<div> Hello World </div>` }
const Error = { template: `<div> Page Not Found</div>` }
const Profile = {
  template: `<div>
    <div v-if='user'>
      Name: {{user.name}}, State: {{user.state}}
    </div>
    <div v-else>
      Unknown User
    </div>
  </div>`,
  data() {
    return {
      users: [
        { id: '1', name: 'Narendra', state: 'UP' },
        { id: '2', name: 'Abhishek', state: 'Delhi' },
      ],
    }
  },
}
```

```

computed: {
  user() {
    user = this.users.find((usr) => {
      return usr.id == this.$route.params.id
    })
    return user
  },
},
}

const routes = [
  { path: '/', component: Home },
  { path: '/profile/:id', component: Profile },
  { path: '*', component: Error },
]

const router = new VueRouter({
  routes,
})

new Vue({
  el: '#app',
  router,
})

```

Suppose the application is running on port 8080. What will be rendered inside router-view for the URL "<http://127.0.0.1:8080/#/profile>"?

Options :

6406531737010. ✔ Page Not Found

6406531737011. ✖ Name: Narendra, State: UP

6406531737012. ✖ Name: Abhishek, State: Delhi

6406531737013. ✖ Unknown User

Question Number : 142 Question Id : 640653521138 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 application with markup “index.html” and javascript file “app.js”.

index.html:

```
<body>
  <div id="app">
    <router-view></router-view>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3/dist/vue-router.js"></script>
  <script src="app.js"></script>
</body>
```

app.js:

```
const Home = { template: `<div> Hello World </div>` }
const DetailsError = { template: `<div> Details not found </div>` }
const PersonalInfo = {
  template: `<div>
    Name: {{user?user.name:'Unknown'}},
    State: {{user?user.state:'Unknown'}}
  </div>`,
  data() {
    return {
      users: [
        { id: '1', name: 'Narendra', state: 'UP' },
        { id: '2', name: 'Abhishek', state: 'Delhi' },
      ],
    }
  },
  computed: {
    user() {
      user = this.users.find((usr) => {
        return usr.id == this.$route.params.id
      })
      return user
    }
  }
}
```



```

    },
  },
}
const ProfessionalInfo = {
  template: `<div> This is professional Info </div>`,
}
const Profile = {
  template: `<div>
<div>Welcome User</div>
<div><router-view></router-view></div>
</div>`,
}

const routes = [
  { path: '/', component: Home },
  {
    path: '/profile/:id',
    component: Profile,
    children: [
      {
        path: 'personal',
        component: PersonalInfo,
      },
      {
        path: 'professional',
        component: ProfessionalInfo,
      },
      { path: '*', component: DetailsError },
    ],
  },
]

const router = new VueRouter({
  routes,
})
new Vue({
  el: '#app',
  router,
})

```

Suppose the application is running on port 8080. What will be rendered inside the router-view component of div element with ID “app”, when a user visits the URL [“http://127.0.0.1:8080/#/profile/2/personal”](http://127.0.0.1:8080/#/profile/2/personal)?

Options :

6406531737022. ✖ Welcome User

Name: Unknown, State: Unknown

6406531737023. ✖ Welcome User

Name: Narendra, State: UP

6406531737024. ✔ Welcome User

Name: Abhishek, State: Delhi

6406531737025. ✖ None of these

Sub-Section Number :	4
Sub-Section Id :	64065373958
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 143 Question Id : 640653521124 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 markup "index.html" and javascript file "app.js".

index.html:

```
<div id = "app">
  <input v-model = "data" />
  <p> Number of refreshes: {{refreshes}} </p>
  <button @click = "do_something"> Click Me</button>
</div>
<script src = "app.js"> </script>
```

app.js:

```
const a = new Vue({
  el : '#app',
  data : {
    data : "",
    refreshes : 0,
  },
  methods: {
    do_something() {
      if (isNaN(this.refreshes)) this.refreshes = 0;
      if (this.data.length % 2) {
        sessionStorage.data = "prefix" + this.data;
        sessionStorage.refreshes = this.refreshes * 2 + 1;
      }
      else {
        sessionStorage.data = this.data + "suffix";
        sessionStorage.refreshes = this.refreshes * 2 - 1;
      }
    }
  },
  mounted : function () {
    if (sessionStorage.data) {
      this.data = "suffix" + sessionStorage.data.slice(1, 5);
      this.refreshes = Number(sessionStorage.refreshes) % 3 - 1;
    }
    else {
      this.data = sessionStorage.data + "prefix";
      this.refreshes = Number(sessionStorage.refreshes) % 3 + 1;
    }
    if (isNaN(this.refreshes)) this.refreshes = 2;
    sessionStorage.data = this.data;
    sessionStorage.refreshes = this.refreshes;
  }
})
```

Say you open the file "index.html" in the browser, and enter the text "study" in the text box shown (after removing the existing text from the input box, if any). After that, you refresh the page twice. What be the text shown in the text input box, and the value of the "refreshes" placeholder, respectively?

Options :

6406531736966. ✖ suffixndef, 0

6406531736967. ✖ suffixndef, -1

6406531736968. ✔ suffixuffi, 0

6406531736969. ✖ suffixuffi, -1

Question Number : 144 Question Id : 640653521129 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 javascript program, and predict the output, if executed.

```
async function test(x) {
  let a = await new Promise(r => r(4 && x || 9)).catch(e => e);
  let b = await new Promise((res, rej) => {
    if (a > 2) res(8 || a && 5);
    else rej(5 && a);
  }).catch(e => e);
  console.log(a, b);
  return (5 && a)
}
test(7).then(
  rej => console.log("Promise rejected with the answer", rej),
  res => console.log("Promise resolved with the answer", res)
).then(data => {
  console.log("New value:", data);
  return "Promise"
}).catch(e => {
  throw new Error("Error 1")
}).finally(data => {
  console.log("New Value:", data);
  return "15"
}).then(data => console.log("New Value:", data)
).finally(() => console.log("End of Program"));
```

Options :

6406531736986. ✖ 7 8

Promise rejected with the answer 8

New value: undefined

New Value: Promise

New Value: 15

End of Program

6406531736987. ✖ End of Program

7 8

Promise rejected with the answer 7

New value: undefined

New Value: undefined

New Value: Promise

6406531736988. ✓ 7 8

Promise rejected with the answer 7

New value: undefined

New Value: undefined

New Value: Promise

End of Program

6406531736989. ✖ 7 8

Promise rejected with the answer 5

New value: undefined

New Value: Promise

New Value: 15

End of Program

Question Number : 145 Question Id : 640653521136 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 app with markup “index.html” and javascript file “app.js”.

index.html:

```
<body>
  <div id="app">
    <router-view></router-view>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3/dist/vue-router.js"></script>
  <script src="app.js"></script>
</body>
```

app.js:

```
const Home = { template: `<div> Hello World </div>` }
const Error = { template: `<div> Page Not Found</div>` }
const Profile = {
  template: `<div>
    <div v-if='user'>
      Name: {{user.name}}, State: {{user.state}}
    </div>
    <div v-else>
      Unknown User
    </div>
  </div>`,
  data() {
    return {
      users: [
        { id: '1', name: 'Narendra', state: 'UP' },
        { id: '2', name: 'Abhishek', state: 'Delhi' },
      ],
    }
  },
}
```

```

computed: {
  user() {
    user = this.users.find((usr) => {
      return usr.id == this.$route.params.id
    })
    return user
  },
},
},
}

const routes = [
  { path: '/', component: Home },
  { path: '/profile/:id', component: Profile },
  { path: '*', component: Error },
]

const router = new VueRouter({
  routes,
})

new Vue({
  el: '#app',
  router,
})

```

Suppose the application is running on port 8080. What will be rendered inside router-view for the URL "<http://127.0.0.1:8080/#/profile/5>"?

Options :

- 6406531737014. ✖ Page Not Found
- 6406531737015. ✖ Name: Narendra, State: UP
- 6406531737016. ✖ Name: Abhishek, State: Delhi
- 6406531737017. ✔ Unknown User

Question Number : 146 Question Id : 640653521137 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 application with markup "index.html" and javascript file "app.js".

index.html:

```
<body>
  <div id="app">
    <router-view></router-view>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
  <script
src="https://unpkg.com/vue-router@3/dist/vue-router.js"></script>
  <script src="app.js"></script>
</body>
```

app.js:

```
const Home = { template: `<div> Hello World </div>` }
const DetailsError = { template: `<div> Details not found </div>` }
const PersonalInfo = {
  template: `<div>
    Name: {{user?user.name:'Unknown'}},
    State: {{user?user.state:'Unknown'}}
  </div>`,
  data() {
    return {
      users: [
        { id: '1', name: 'Narendra', state: 'UP' },
        { id: '2', name: 'Abhishek', state: 'Delhi' },
      ],
    }
  },
  computed: {
    user() {
      user = this.users.find((usr) => {
        return usr.id == this.$route.params.id
      })
    }
  }
}
```



```

    })
    return user
  },
},
}

const ProfessionalInfo = {
  template: `<div> This is professional Info </div>`,
}

const Profile = {
  template: `<div>
    <div>Welcome User</div>
    <div><router-view></router-view></div>
  </div>`,
}

const routes = [
  { path: '/', component: Home },
  {
    path: '/profile/:id',
    component: Profile,
    children: [
      {
        path: 'personal',
        component: PersonalInfo,
      },
      {
        path: 'professional',
        component: ProfessionalInfo,
      },
      { path: '*', component: DetailsError },
    ],
  },
]

const router = new VueRouter({
  routes,
})

new Vue({
  el: '#app',
  router,
})

```

Suppose the application is running on port 8080. What will be rendered inside the router-view component of div element with ID “app”, when the user visits the URL [“http://127.0.0.1:8080/#/profile/3/personal”](http://127.0.0.1:8080/#/profile/3/personal)?

Options :

6406531737018. ✔ Welcome User

Name: Unknown, State: Unknown

6406531737019. ✖ Welcome User

Name: Narendra, State: UP

6406531737020. ✖ Welcome User

Name: Abhishek, State: Delhi

6406531737021. ✖ None of these

Sub-Section Number :	5
Sub-Section Id :	64065373959
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 147 Question Id : 640653521123 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are false regarding promise chain in javascript language?

Options :

- 6406531736962. ✖ A promise chain may consist of a number of “then” blocks.
- 6406531736963. ✔ A “finally” block always comes at the end of the promise chain.
- 6406531736964. ✔ Every “catch” block must always be preceded by a “then” block.
- 6406531736965. ✖ The “finally” block always gets executed, irrespective of the promise outcome.

Sub-Section Number :	6
Sub-Section Id :	64065373960
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 148 Question Id : 640653521127 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are true regarding web storage APIs?

Options :

- 6406531736978. ✔ The data saved in local storage has no expiration time, unlike session storage

(except in private browsing).

6406531736979. ✖ Both the local storage and session storage return the same object for site loaded over HTTP and HTTPS.

6406531736980. ✖ The data saved in local storage is synced across the devices.

6406531736981. ✔ The data saved in session storage gets cleared as soon as the page session ends.

MLT

Section Id :	64065333938
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	15
Number of Questions to be attempted :	15
Section Marks :	100
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 :	64065373961
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 149 Question Id : 640653521139 Question Type : MCQ Is Question

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

Correct Marks : 0