

Question Label : Multiple Choice Question

What will be output on the terminal
for the command

```
python file.py course 1?
```

Options :

6406532733824. ✖ `No course found!`

6406532733825. ✔ `course found App Dev I`

6406532733826. ✖ `course found App Dev I
course found App Dev II
course found App Dev III
course found DevOps`

6406532733827. ✖ `IndexError: list index out of range`

AppDev2

Section Id :	64065356695
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	33
Number of Questions to be attempted :	33
Section Marks :	100
Display Number Panel :	Yes

Section Negative Marks :	0
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 :	640653118937
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 285 Question Id : 640653816158 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 :

6406532733916.  YES

6406532733917.  NO

Sub-Section Number :	2
Sub-Section Id :	640653118938
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 286 Question Id : 640653816159 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 following Script embedded in an HTML document.

```
let langName = 'Python';  
function showLanguage() {  
  let langName = "JavaScript";  
  let message = 'Learn ' + langName;  
  console.log(message);  
}  
  
let message = 'Learn ' + langName;  
console.log(message)  
console.log(showLanguage());
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

6406532733918. ✖ ☐ Learn JavaScript
☐ Learn Python

6406532733919. ✖ ☐ Learn JavaScript
☐ Learn JavaScript
☐ undefined

6406532733920. ✖ ☐ Learn Python
☐ Learn JavaScript

6406532733921. ✔ ☐ Learn Python
☐ Learn JavaScript
☐ undefined

Question Number : 287 Question Id : 640653816160 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 following Script embedded in an HTML document.

```
let x = [1, 'p', a => a + 1];  
  
for (const i = 0; i < x.length; i++) {  
    console.log(i, x[i], typeof(x[i]))  
}
```

Which of the following statement is correct, if the HTML document is rendered using a browser?

Options :

6406532733922. ✖ It will throw error on the console for the very first iteration.

6406532733923. ✔ It will display the index, value and type of the value for the first item of array and then will throw error for the next value.

6406532733924. ✖ It will display the indices, values and types of all the items of the array.

6406532733925. ✖ None of these.

Question Number : 288 Question Id : 640653816166 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

Assuming the URL gives a valid json response, what will be the output in the console?

```
fetch('https://study.iitm.ac.in/api/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));

console.log('End of script');
```

Options :

6406532733946. ✖ The JSON data from the specified URL followed by "End of script".

6406532733947. ✔ "End of script" will be logged first, followed by the JSON data.

6406532733948. ✖ An error will be logged, and the script will terminate.

6406532733949. ✖ "End of script" will be logged, but the JSON data retrieval will fail.

Question Number : 289 Question Id : 640653816172 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 JavaScript and its behavior?

Options :

6406532733970. ✖ The language supports both first class and higher order functions.

6406532733971. ✔ The language supports higher order functions, but not first class functions.

6406532733972. ✖ A variable declared without keywords "var", "let" or "const" is not hoisted at all.

6406532733973. ✖ The variables declared using keywords "let" and "const" are also hoisted.

Question Number : 290 Question Id : 640653816174 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

Where is system state commonly stored in a web application?

Options :

6406532733978. ✖ Client Side local storage

6406532733979. ✖ In Memory Cache

6406532733980. ✖ Session Cookies

6406532733981. ✔ Server Side Database

Question Number : 291 Question Id : 640653816175 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 technology is commonly used to manage application state in a Vue.js application?

Options :

6406532733982. ✖ Vue.js

6406532733983. ✖ React.js

6406532733984. ✔ Vuex

6406532733985. ✖ Redux

6406532733986. ✖ Express.js

6406532733987. ✖ Angular

Sub-Section Number : 3

Sub-Section Id : 640653118939

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 292 Question Id : 640653816161 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

Consider the following Script embedded in an HTML document.

```
let myQual = {  
  degree: "B-tech",  
  college: "IIT Delhi",  
  get qualification(){  
    return `degree: ${this.degree}, college: ${this.college}`  
  },  
  set qualification(q){  
    let components = q.split(' ');  
    this.degree = components[0];  
    this.college = components[1];  
  }  
}
```

Which of the following statements will throw an error on the console when the HTML document is rendered using a browser?

Options :

6406532733926. ✓ `console.log(myQual.qualification())`

6406532733927. ✗ `console.log(myQual.qualification)`

6406532733928. ✗ `console.log(myQual.qualification = 'M-tech IIT-Madras')`

6406532733929. ✓ `console.log(myQual.qualification('M-tech IIT-Madras'))`

Question Number : 293 Question Id : 640653816169 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

Consider the following two HTML documents and the Vue app created in script.js file and select the correct option(s).

index1.html

```
<body>
  <div id="app">
    <h1 v-if="result">Hello am I visible?</h1>
  </div>
  <script src="./script.js"></script>
</body>
```

index2.html

```
<body>
  <div id="app">
    <h1 v-show="final">Hello am I visible?</h1>
  </div>
  <script src="./script.js"></script>
</body>
```

script.js

```
var app = new Vue({
  el: '#app',
  data: {
    result: false,
    final: false
  }
})
```

Options :

6406532733958. ✖ The raw HTML created with index1.html and index2.html will be exactly the same for the current state of data.

6406532733959. ✔ When the directive v-if is falsy, it removes the entire element from the DOM.

6406532733960. ✖ When the directive v-show is falsy, it removes the entire element from the DOM.

The raw HTML created with `index1.html` and `index2.html` will be exactly the same when the values of both "result" and "final" are set to true.
6406532733961. ✓

Question Number : 294 Question Id : 640653816170 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

Consider the following Vue application and the rendered HTML output below.

Filename: script.js

```
var app = new Vue({
  el: '#app',
  data: {
    items:[
      {id: 1, name:"mad 1"},
      {id: 2, name:"mad 2"},
      {id: 3, name:"mad 3"}
    ]
  }
})
```

Rendered output:

mad 1

mad 2

mad 3

What should be the content of `<div id = "app"></div>` in index.html that generates the given rendered output?

Options :

6406532733962. ✓

```
<h2 v-for="item in items">{{item.name}}</h2>
```

6406532733963. ✖

```
<h2 v-for="item in items" {{item.id}}>{{item.name}}</h2>
```

6406532733964. ✔

```
<h2 v-for="item in items" :id="item.id">{{item.name}}</h2>
```

6406532733965. ✖

```
<h2 v-for="items" :id="item.id">item.name</h2>
```

Question Number : 295 Question Id : 640653816180 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 HTTP status code(s) is/are commonly used in RESTful APIs for successful responses?

Options :

6406532734005. ✔ 200

6406532734006. ✔ 201

6406532734007. ✖ 400

6406532734008. ✖ 404

6406532734009. ✖ 500

Question Number : 296 Question Id : 640653816186 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 is/are potential benefit(s) of using caching in web applications?

Options :

- 6406532734030. ✔ Reduced latency and faster response times
- 6406532734031. ✔ Lower server load and reduced bandwidth usage
- 6406532734032. ✔ Improved scalability and better handling of traffic spikes
- 6406532734033. ✖ Enhanced security and protection against DDoS attacks

Sub-Section Number :	4
Sub-Section Id :	640653118940
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 297 Question Id : 640653816162 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 Script embedded in an HTML document.

```
class Twowheeler {
  constructor(name){
    this.name = name;
    this.gear = 4;
    this.seating = 2;
  }
  get description(){
    return `${this.name} has ${this.gear} gear and has
    ${this.engine} engine.`
  }
}

class Moped extends Twowheeler {
  constructor(name){
    super(name);
    this.engine = '4 stroke'
  }
}

let myBike = new Twowheeler("Discover")
let myDrive = new Moped("Activa")
console.log(myDrive.description)
console.log(myBike.description)
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

6406532733930. ✖

```
Discover has 4 gear and has 4 stroke engine.
Activa has 4 gear and has 4 stroke engine.
```

6406532733931. ✖

```
Activa has 4 gear and has undefined engine.
Discover has 4 gear and has 4 stroke engine.
```

6406532733932. ✔

```
Activa has 4 gear and has 4 stroke engine.
Discover has 4 gear and has undefined engine.
```

Discover has 4 gear and has undefined engine.
Activa has 4 gear and has undefined engine.

6406532733933. ✖

Question Number : 298 Question Id : 640653816164 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 Script embedded in an HTML document.

```
function parentFunc(name) {  
  return {  
    sayHello: () => "Hi! " + name,  
    sayBye: () => "Bye! " + name,  
    changeName: (newName) => {  
      name = newName;  
    },  
  };  
}  
const arpan = parentFunc("Arpan");  
const beli = parentFunc("Beli");  
  
console.log(arpan.sayHello());  
console.log(beli.sayHello());  
arpan.changeName("Beli");  
console.log(arpan.sayBye());
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

Hi! Arpan
Bye! Arpan

6406532733938. ✖

6406532733939. ✔

```
Hi! Arpan  
Hi! Beli  
Bye! Beli
```

```
Hi! Arpan  
reference Error
```

6406532733940. ✖

6406532733941. ✖ Reference Error

Question Number : 299 Question Id : 640653816165 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

What will be the text displayed in a web page if the Vue app given below is running in a development mode?

Filename: index.html

```
<div id="app1">
  <h1>{{ title }}</h1>
  <h4>{{ greetings() }}</h4>
</div>
<div id="app2">
  <h1>{{ title }}</h1>
  <h4>{{ greetings }}</h4>
</div>
```

Filename: app.js

```
new Vue({
  el: "#app1",
  data: { title: "App No 1" },
  computed: {
    greetings() {
      return "hello from " + this.title;
    },
  },
});

new Vue({
  el: "#app2",
  data: { title: "App No 2" },
  computed: {
    greetings() {
      return "hello from " + this.title;
    },
  },
});
```

Options :

App No 1

hello from App No 1

App No 2

hello from App No 2

6406532733942. ✖

6406532733943. ✔

App No 2

hello from App No 2

App No 1

App No 2

6406532733944. ✖

6406532733945. ✖ Blank page

Question Number : 300 Question Id : 640653816168 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 Script embedded in an HTML document.

```
var var1 = 25;
var var2 = 35;

const myObj = {
  var2: 45,
  var1: 35,
  ObjFunc: function(var3){
    let var4 = var3**2;
    return 10 + this.var2 + var4;
  }
}

let m = myObj.ObjFunc
console.log(m.bind()(5))
console.log(m.call(myObj,6))
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

6406532733954. ✖

70
81

70
91

6406532733955. ✓

80
81

6406532733956. ✖

80
91

6406532733957. ✖

Question Number : 301 Question Id : 640653816178 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

What does the “finally” method do in promises?

Options :

6406532733997. ✖ The method is executed when the promise is fulfilled.

6406532733998. ✖ The method is executed when the promise is rejected.

6406532733999. ✓ The method is executed regardless of whether the promise is fulfilled or rejected.

6406532734000. ✖ The method can not be used with promises at all.

Question Number : 302 Question Id : 640653816181 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

Which of the following statements regarding Celery task execution is correct?

Options :

6406532734010. ✖ Celery tasks are executed synchronously by default.

6406532734011. ✖ Celery tasks can only be executed on the same machine where the flask application is running.

6406532734012. ✔ Celery tasks are executed asynchronously by default.

6406532734013. ✖ Celery tasks can only be executed once they are defined as part of a flask application.

Question Number : 303 Question Id : 640653816182 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

What is the primary objective of using a message broker?

Options :

6406532734014. ✖ Eliminates the need for network communication

6406532734015. ✔ Decouples producers and consumers

6406532734016. ✖ Decreases CPU utilization

6406532734017. ✖ Reduces the need for data serialization

Question Number : 304 Question Id : 640653816183 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

In a message broker system, what does the term "topic" refer to?

Options :

6406532734018. ✖ The physical location where messages are stored.

6406532734019. ✖ A unique identifier for a message.

6406532734020. ✖ The process of encrypting messages.

6406532734021. ✔ A channel to which messages are published.

Question Number : 305 Question Id : 640653816184 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

What is the primary advantage of using server-sent events over traditional polling?

Options :

6406532734022. ✖ Server-sent events support bidirectional communication.

6406532734023. ✖ Server-sent events can handle larger payloads.

6406532734024. ✔ Server-sent events reduce server load by eliminating the need for frequent polling.

6406532734025. ✖ Server-sent events provide lower latency compared to polling.

Question Number : 306 Question Id : 640653816185 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

Which of the following protocols is commonly used for implementing polling?

Options :

6406532734026. ✔ HTTP

6406532734027. ✖ Web Sockets

6406532734028. ✖ FTP

6406532734029. ✖ SMTP

Question Number : 307 Question Id : 640653816187 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

What is the primary purpose of CORS in web security?

Options :

6406532734034. ✖ To prevent unauthorized access to confidential data

6406532734035. ✖ To mitigate cross-site scripting (XSS) attacks

6406532734036. ✔ To enable controlled access to resources from different origins

6406532734037. ✖ To protect against SQL injection attacks

Sub-Section Number :	5
Sub-Section Id :	640653118941
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 308 Question Id : 640653816163 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 Script embedded in an HTML document.

```
let Obj1 = {  
  subject: 'Mechanics',  
  stream: 'Physics'  
}  
  
let Obj2 = Obj1;  
let Obj3 = {};  
  
for (let key in Obj1){  
  Obj3[key] = Obj1[key];  
}  
  
Obj2.subject = 'Thermodynamics'  
Obj3.stream = 'Chemistry'  
  
console.log(Obj1)
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

```
{  
  subject: 'Thermodynamics',  
  stream: 'Physics'  
}
```

6406532733934. ✓

```
{  
  subject: 'Mechanics',  
  stream: 'Chemistry'  
}
```

6406532733935. ✖

```
{  
  subject: 'Mechanics',  
  stream: 'Physics'  
}
```

6406532733936. ✖

```
{  
  subject: 'Thermodynamics',  
  stream: 'Chemistry'  
}
```

6406532733937. ✖

Question Number : 309 Question Id : 640653816167 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 Script embedded in an HTML document.

```
var var1 = 25;

const exObj = {
  var2: 45,
  var1: 35,
  inObj : {
    var1: 45,
    inObjFunc: ()=>{
      return "Value is " + this.var1;
    }
  },
  exObjFunc: function(){
    let var2 = 10;
    return "Value is " + this.var2;
  }
}

let x = exObj.inObj
console.log(x.inObjFunc())
console.log(exObj.exObjFunc())
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

6406532733950. ✖

Value is 35
Value is 10

6406532733951. ✖

Value is 35
Value is 45

6406532733952. ✖

Value is 25
Value is 10

6406532733953. ✔

```
Value is 25  
Value is 45
```

Question Number : 310 Question Id : 640653816171 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 Script "script.js" embedded in an HTML document, "index.html" and an external script "app.js" given below.

Filename: index.html

```
<html lang="en">
<head></head>
<body>
  <script src="./script.js"></script>
</body>
</html>
```

Filename: script.js

```
function loadScript(src, cbf){
  let script = document.createElement('script');
  script.src = src;
  script.onload = () => cbf();
  document.head.append(script);
}

let cbf = function(){
  console.log("callback function executed here.");
  extFunc()
}

loadScript("./support_m6.js", cbf)
extFunc()
```

Filename: app.js

```
console.log("Now I am loaded, let's support")

function extFunc(){
  console.log("loaded and run from external script")
}
```

What will be the output on console, if the HTML document is rendered using a browser?

Options :

```
Now I am loaded, let's support
callback function executed here.
loaded and run from external script
loaded and run from external script
```

6406532733966. ✖

6406532733967. ✖

```
loaded and run from external script
Now I am loaded, let's support
callback function executed here.
loaded and run from external script
```

6406532733968. ✖

```
Now I am loaded, let's support
callback function executed here.
loaded and run from external script
Uncaught ReferenceError: extFunc is not defined
```

6406532733969. ✔

```
Uncaught ReferenceError: extFunc is not defined
Now I am loaded, let's support
callback function executed here.
loaded and run from external script
```

Question Number : 311 Question Id : 640653816173 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.

```
Array.prototype.double = function (arr) {  
  const res = [];  
  for (let i=0; i<placeholder.length; i++)  
    res.push(placeholder[i] * 2);  
  return res;  
}  
  
const arr = [2, 3, 7, 8];  
console.log(arr.double());
```

What of the following can be used in place of “placeholder” in the above program so that the program yields the below output array?
[4, 6, 14, 16]

Options :

6406532733974. ✖ arr

6406532733975. ✖ Array

6406532733976. ✔ this

6406532733977. ✖ The program cannot yield such an output, and is implemented in a wrong way.

Question Number : 312 Question Id : 640653816177 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.

```
const fetchData = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url.includes("success")) {
        resolve(`Data fetched successfully from: ${url}`);
      } else {
        reject(`Error fetching data: ${url}`);
      }
    }, 1000);
  });
};

const processData = (data) => {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve(`Processed data: ${data.toUpperCase()}`);
    }, 500);
  });
};

const handleError = (error) => {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve(`Handled Error: ${error}`);
    }, 300);
  });
};

fetchData("https://example.com/success")
  .then(processData)
  .then(fetchData)
  .then(handleError)
  .then((result) => {
    console.log(result);
  })
  .catch((error) => {
    console.error(error);
  });
```

What will be the output of the above program, if executed?

Options :

Handled Error: Processed data: DATA FETCHED SUCCESSFULLY FROM:
6406532733992. ✖ HTTPS://EXAMPLE.COM/SUCCESS

Error fetching data: Processed data: DATA FETCHED SUCCESSFULLY FROM:
6406532733993. ✔ HTTPS://EXAMPLE.COM/SUCCESS

6406532733994. ✖ Handled Error: ERROR FETCHING DATA FROM: UNDEFINED

6406532733995. ✖ Handled Error: ERROR FETCHING DATA FROM: [HTTPS://EXAMPLE.COM/SUCCESS](https://example.com/success)

6406532733996. ✖ Error fetching data: Processed data: UNDEFINED

Question Number : 313 Question Id : 640653816179 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 Vuex module definition for managing a shopping cart.

```
const cartModule = {
  state: {
    items: []
  },
  mutations: {
    addItem(state, item) {
      state.items.push(item);
    },
  },
  actions: {
    async addToCart({ commit }, item) {
      // TODO: Implement adding item to cart
    },
  },
}
```

Which of the following options correctly completes the code for the “addToCart” action function?

Options :

```
addToCart({ commit }, item) {
  commit('addItem', item);
}
```

6406532734001. ✔

```
addToCart({ state, commit }, item) {  
  const existingItem = state.items.find(i => i.id === item.id);  
  if (existingItem) {  
    commit('addItem', existingItem);  
  } else {  
    commit('addItem', item);  
  }  
}
```

6406532734002. ✖

```
addToCart({ commit }, item) {  
  commit('addItem', { ...item });  
}
```

6406532734003. ✖

6406532734004. ✖ All of these

Question Number : 314 Question Id : 640653816188 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 = "subject" @input = "compute_marks">
  <p> {{marks}} </p>
</div>
<script scr = "app.js"></script>
```

app.js:

```
new Vue({
  el : "#app",
  data : {
    subject : "AppDev",
    marks : 50,
  },

  mounted () {
    this.subject = "AppDev";
    this.marks = 50;

    if (localStorage.marks) {
      this.subject += "1";
      this.marks = parseInt(localStorage.marks) + 20;
      localStorage.marks = this.marks;
      localStorage.subject = this.subject;
    }
    else {
      this.subject += "2";
      this.marks += 20;
    }
  },

  methods : {
    compute_marks() {
      localStorage.setItem("subject", this.subject);
      localStorage.setItem("marks", this.marks + 10);
    }
  }
})
```

Suppose you open "index.html" file in a browser, and type the text "EndTermExam" in the text box shown (after removing the previous text, if any), and hard refresh the page thrice, without clicking anywhere. What will be the value shown in the text box, and the "marks" placeholder, respectively?

Options :

6406532734038. ✖ AppDev1, 120

6406532734039. ✖ AppDev2, 120

6406532734040. ✔ AppDev1, 140

6406532734041. ✖ AppDev2, 140

Question Number : 315 Question Id : 640653816189 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">
  <router-link :to="{ name: 'user', params: { id: 'Jyoti' }}">User
  Profile</router-link>
  <router-view></router-view>
</div>
<script src="app.js"></script>
```

app.js:

```
const UserProfile = {
  template: `
    <div>
      <h1>User Profile</h1>
      <p>User ID: {{ id }}</p>
    </div>
  `,
  props: ['id']
};

const routes = [
  { path: '/user/:id', name: 'user', component: UserProfile, props: true }
];

const router = new VueRouter({
  routes
});

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

Based on this setup, what will be displayed in the browser when the user clicks on the "User Profile" link?

Options :

6406532734042. ✖ An error indicating that the id prop is not defined.

6406532734043. ✔ User Profile

User ID: Jyoti

6406532734044. ✖ An error indicating that the data object is not defined.

6406532734045. ✖ An error indicating that the route does not exist.

6406532734046. ✖ User Profile

User ID:

Sub-Section Number :	6
Sub-Section Id :	640653118942
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 316 Question Id : 640653816176 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 is/are built in Vue.js lifecycle hook(s)?

Options :

6406532733988. ✖ beforeLoad

6406532733989. ✖ preUpdate

6406532733990. ✔ mounted

6406532733991. ✔ updated

Question Number : 317 Question Id : 640653816190 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 JavaScript method(s) is/are used to remove data from the session storage?

Options :

6406532734047. ✔ clear()

6406532734048. ✔ removeItem()

6406532734049. ✖ deleteItem()

6406532734050. ✖ unSet()

MLT

Section Id :	64065356696
Section Number :	11
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
Section Negative Marks :	0
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 :	640653118943
Question Shuffling Allowed :	No