Options :

6406532867113. ✖
```
public static void printMostImpressiveAct(<?> performers)
```

6406532867114. ✔
```
public static <T extends Performer> void
   printMostImpressiveAct(T[] performers)
```

6406532867115. ✖
```
public static <T extends Magician> void
   printMostImpressiveAct(T[] performers)
```

6406532867116. ✔
```
public static void printMostImpressiveAct(Performer[] performers)
```

AppDev2

Section Id :	64065360929
Section Number :	7
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 :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653126859
Question Shuffling Allowed :	No

Question Number : 105 Question Id : 640653852439 Question Type : MCQ

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 :

6406532867157. ✔ YES

6406532867158. ✖ NO

Sub-Section Number : 2

Sub-Section Id : 640653126860

Question Shuffling Allowed : Yes

Question Number : 106 Question Id : 640653852440 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the below JavaScript program.

```
const first = "Hello 3";

const obj1 = {
    first: "Hello 1",
    secondFunc: function () {
        console.log(this.first);
    }
}

const obj2 = {
    first: "Hello 2",
    secondFunc: function () {
        console.log(this.first);
        this.secondFunc();
    }
}

obj1.secondFunc.apply(obj2, [2, 5, 9]);
```

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

Options :

6406532867159. ✖ Hello 1

6406532867160. ✖ Hello 1
Hello 2

6406532867161. ✔ Hello 2

6406532867162. ✖ The program will result in an infinite loop

Question Number : 107 Question Id : 640653852442 Question Type : MCQ
Correct Marks : 2
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 #header>
            This is header content
        </template>

        <template #footer>
            This is footer content
        </template>

        <p> This is some content </p>
    </my-comp>
</div>
<script src = "app.js"> </script>
```

app.js:

```
Vue.component("myComp", {
    template : `<div>
                    <p>
                        <slot></slot>
                    </p>
                    <p>
                        <slot name="footer"></slot>
                    </p>
                </div>
                `,
})

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

Suppose you open the "index.html" file in a browser. What will be rendered by the browser?

Options :

6406532867167. ✖ This is header content
This is footer content

This is some content
6406532867168. ✖ This is footer content
This is some content
6406532867169. ✖ This is some content
6406532867170. ✖ This is some content
This is header content
6406532867171. ✔ This is some content
This is footer content

Sub-Section Number :	3
Sub-Section Id :	640653126861
Question Shuffling Allowed :	Yes

Question Number : 108 Question Id : 640653852441 Question Type : MCQ
Correct Marks : 3
Question Label : Multiple Choice Question

Consider the following application with markup index.html and script app.js.

index.html:

```
<body>
    <div id="app">
        <router-view></router-view>
    </div>
</body>
```

app.js:

```
const Header = {
  template: `<div>This is header <router-view /> </div>`,
}
const Error = {
  template: '<div> 404 Page not found</div>',
}
const Profile = {
  template: `<div> Welcome to profile page</div>`,
}
const routes = [
  {
    path: '/',
    component: Header,
    children: [
      { path: 'profile', component: Profile },
      { path: '*', component: Error },
    ],
  },
]
const router = new VueRouter({
  routes,
})
new Vue({
  el: '#app',
  router,
})
```

Suppose the application is running on http://127.0.0.1:8080. What will be rendered in the browser?

Options :

6406532867163. ✔ This is header
404 Page not found

6406532867164. ✖ 404 Page not found
This is header

6406532867165. ✖ Welcome to profile page
This is header

6406532867166. ✖ This is header
Welcome to profile page

Question Number : 109 Question Id : 640653852444 Question Type : MCQ
Correct Marks : 3
Question Label : Multiple Choice Question

Consider the below JavaScript program.

```
new Promise((error, pass) => {
    if (5 === "5") error(5)
    else pass(8)
}).
then(d => {
    console.log("Checkpoint 4", d);
    throw new Error(20);
    return d * 5;
})
.then(d => {
    console.log("Checkpoint 2", d);
    return d;
}, d => {
    console.log("Checkpoint 5", d.message);
    return d.message * 2;
}).catch(e => {
    console.log("Checkpoint 3", e.message);
    return e.message * 2;
}).finally(d => {
    console.log("Checkpoint 1", d);
    return d * 5;
}).then(d => {
    console.log("Checkpoint 6", d);
    return d * 5;
})
```

What will be the output of the above program?

Options :

6406532867176. ✖ Checkpoint 4 5
Checkpoint 5 20
Checkpoint 1 undefined
Checkpoint 6 40

6406532867177. ✖ Checkpoint 4 5
Checkpoint 3 20
Checkpoint 1 undefined
Checkpoint 6 NaN

6406532867178. ✔ Checkpoint 5 undefined
Checkpoint 1 undefined
Checkpoint 6 NaN

6406532867179. ✖ Checkpoint 5 undefined
Checkpoint 1 16
Checkpoint 6 80

6406532867180. ✖ Checkpoint 5 undefined
Checkpoint 1 undefined
Checkpoint 6 16

Question Number : 110 Question Id : 640653852445 Question Type : MCQ
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">
    <input v-model = "phrase" @input = "compute_marks">
    <p> {{marks}} </p>
</div>
<script scr = "app.js"></script>
```

app.js:

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

    mounted() {
        this.phrase = "Bravo";
        this.marks = 50;

        if (localStorage.marks) {
            this.phrase += "2";
            this.marks = parseInt(localStorage.marks) + 20;
        }
        else {
            this.phrase += "1";
            this.marks += 10;
        }
    },

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

Suppose you open “index.html” file in a browser, and type the text “Hurray” in the text box shown (after removing the previous text, if any), and hard refresh the page 7 times, without clicking anywhere. What will be the value shown in the text box, and the “marks” placeholder, respectively?

Tip: The “oninput” event occurs immediately after the value of an element has changed.

Options :

6406532867181. ✖ Bravo1, 70

6406532867182. ✔ Bravo2, 90

6406532867183. ✖ Bravo1, 60

6406532867184. ✖ Bravo2, 70

6406532867185. ✖ Bravo1, 80

Question Number : 111 Question Id : 640653852447 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

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

```
async function exam () {
    let x = await new Promise((res, rej) => res(2 || 0));
    console.log("Statement 1");
    let y = await new Promise((res, rej) => res(0 && 2));
    console.log("Statement 2", "gives result as", x + y);
}
exam();
console.log("Statement 3");
```

Options :

6406532867190. ✖ Statement 3
Statement 1
Statement 2 gives result as 4

6406532867191. ✔ Statement 3
Statement 1
Statement 2 gives result as 2

6406532867192. ✖ Statement 3
Statement 1
Statement 2 gives result as 0

6406532867193. ✖ Statement 1
Statement 3
Statement 2 gives result as 4

6406532867194. ✖ Statement 1
Statement 3
Statement 2 gives result as 2

Question Number : 112 Question Id : 640653852448 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Vue application with markup index.html and JavaScript file app.js.

File: index.html

```
<div id="app"></div>
<script src="script.js"></script>
```

File: script.js

```
new Vue({
    el: '#app',
    template: `<div>
                    Status: {{message}}<br>
                    ETA: {{ETA}}
                </div>`,
    data: {
      message: "",
      ETA: 10,
    },
    beforeCreate() {
        this.message += "Creating, "
        this.ETA -= 1
    },
    created() {
        this.message += "Created, "
        this.ETA -= 1
    },
    beforeMount() {
        this.message += "Mounting, "
        this.ETA -= 1
    },
    mounted() {
        this.message += "Mounted"
        this.ETA -= 1
    },
  })
```

Suppose the application is running on `http://127.0.0.1:8080`. What will be rendered by the browser?

Options :

6406532867195. ✖ Status: Creating, Created, Mounting, Mounted
ETA: 6

6406532867196. ✔ Status: Created, Mounting, Mounted
ETA: 7

6406532867197. ✖ Status: Mounting, Mounted
ETA: 8

6406532867198. ✖ Status: Mounted
ETA: 9

Question Number : 113 Question Id : 640653852451 Question Type : MCQ
Correct Marks : 3
Question Label : Multiple Choice Question
Consider the following JavaScript code.

```
const library = {
  name: "Central Library",
  books: [
    { title: "Book A", author: "Author A" },
    { title: "Book B", author: "Author B" },
    { title: "Book C", author: "Author C" },
  ],
  location: "IIT Madras",
};

let propertyList = "";
let bookDetails = "";

for (let key in library) {
  if (key === "books") {
    for (let book of library[key]) {
      bookDetails += `${book.title} by ${book.author}, `;
    }
  } else {
    propertyList += key + " ";
  }
}

console.log(propertyList);
console.log(bookDetails);
```

What will be the output of the above code snippet on the console of the browser?
Options :

6406532867207. ✖

```
name books location
Book A by Author A, Book B by Author B, Book C by Author C,
```

6406532867208. ✖

```
Central Library IIT Madras
Book A by Author A, Book B by Author B, Book C by Author C,
```

6406532867209. ✔

```
name location
Book A by Author A, Book B by Author B, Book C by Author C,
```

6406532867210. ✖

```
Central IIT
Book A by Author A, Book B by Author B, Book C by Author C,
```

Question Number : 114 Question Id : 640653852452 Question Type : MCQ
Correct Marks : 3
Question Label : Multiple Choice Question

Consider the following HTML snippet that uses Vue 2 using CDN in the script.js file

Filename: index.html

```
<div id="app">
    <p>Counter: {{ counter }}</p>
    <p>Computed Value: {{ computedValue }}</p>
</div>
```

Filename: script.js

```
new Vue({
    el: "#app",
    data: {
       counter: 0,
    },
    computed: {
      computedValue() {
        return this.counter * 2;
      },
    },
    watch: {
      counter(newVal, oldVal) {
        console.log(`Counter changed from ${oldVal} to ${newVal}`);
      },
    },
    mounted() {
      setInterval(() => {
        this.counter++;
      }, 1000);
    },
});
```

What will be displayed in the console?

Options :

6406532867211. ✖ Counter changed from 0 to 2, Counter changed from 2 to 4, Counter changed from 4 to 6, ...

6406532867212. ✔ Counter changed from 0 to 1, Counter changed from 1 to 2, Counter changed

from 2 to 3, ...

6406532867213. ✖ Counter changed from 1 to 3, Counter changed from 3 to 6, Counter changed from 6 to 12, ...

6406532867214. ✖ error message

Question Number : 115 Question Id : 640653852454 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Suppose you are organizing a felicitation program for the students with exemplary performance in the BS program. You have the scorecard of all the students. The administration has decided to shortlist the students based on the following 2 criteria:

1. The student must have an "S" in their MAD II project
2. The overall CGPA of the student must be greater than 8

Fill in the ***code1*** & ***code2,*** which can be used in Vuex Store to update the "awardees" state variable with the objects of those students who satisfy the above-mentioned criteria. The objects to be inserted in the "awardees" state variable must only have the student ID and a random token generated in the "send_task" action function.

```
const store = new Vuex.Store({
    state : {
        bs_program : [
            {
                student_id : 'stu1001',
                mad2_grade : 'S',
                cgpa : 8.35
            },
            {
                student_id : 'stu1002',
                mad2_grade : 'A',
                cgpa : 8.78
            },
            {
                student_id : 'stu1003',
                mad2_grade : 'S',
                cgpa : 9.1
            },
            {
                student_id : 'stu1004',
                mad2_grade : 'S',
                cgpa : 7.44
            },
        ],
        awardees : []
    },

    mutations : {
        update_final(state, payload) {

            for (student of state.bs_program)
                code2
        }
    },
    actions : {
        send_task : function (context) {
            const token = get_token(); // generates a random token
            code1
        }
    }
})
```

Options :

6406532867219. ✖

```
code1: this.$store.commit("update_final", {“min” : 8, “token” : token});
code2: if (student.mad2_grade != “S” && student.cgpa > 8) {
          const obj = {};
          obj.student_id = student.student_id;
          obj.token = payload.token
          state.awardees.push(student)
       }
```

6406532867220. ✖

```
code1: context.commit("update_final", 8, token);
code2: if (mad2_grade != “S” && cgpa > 8){
          awardees.push(company)
       }
```

6406532867221. ✖

```
code1: this.$store.commit("update_final", 8, token);
code2: if (mad2_grade != “S” && cgpa > 8){
          awardees.push(company)
       }
```

6406532867222. ✔

```
code1: context.commit("update_final", {“min” : 8, “token” : token});
code2: if (student.mad2_grade != “S” && student.cgpa > 8) {
          const obj = {};
          obj.student_id = student.student_id;
          obj.token = payload.token
          state.awardees.push(student)
       }
```

Sub-Section Number : 4

Sub-Section Id : 640653126862

Question Shuffling Allowed : Yes

Question Number : 116 Question Id : 640653852443 Question Type : MSQ

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the below Vue class binding and select the most appropriate option(s).

```
<div v-bind:class="[isClassA ? classA : '', classB]"></div>
```

Options :

6406532867172. ✖ The classes, namely “classA” and “classB” will always be applied to the div element.

6406532867173. ✔ The class, namely “classB” will always be applied to the div element.

6406532867174. ✔ The class, namely “classA” will only be applied to the div element, if the

variable “isClassA” evaluates to true.

6406532867175. ✖ The class, namely “classB” will only be applied to the div element, if no variable with name “isClassA” exists.

Question Number : 117 Question Id : 640653852446 Question Type : MSQ

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are incorrect regarding Vuex?

Options :

6406532867186. ✖ A Vuex store provides a single source of truth that can drive the application.

6406532867187. ✔ It provides a variable named “this.$vuexstore” to allow components to access the store data.

6406532867188. ✔ A component cannot have its own local state, if the application uses Vuex.

6406532867189. ✖ The mutations and actions are some constructs of a Vuex store.

Sub-Section Number : 5

Sub-Section Id : 640653126863

Question Shuffling Allowed : Yes

Question Number : 118 Question Id : 640653852449 Question Type : MCQ

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the following HTML, with Vue 2 CDN attached to it.

Filename: index.html

```
<div id="app">
  <p>{{ message }}</p>
  <button @click="changeMessage">Change Message</button>
  <button @click="changeCount">Change Count</button>
</div>
```

Filename: script.js

```
<script>
  new Vue({
    el: "#app",
    data: {
      message: "Hello, Vue!",
      count: 0,
    },
    methods: {
      changeMessage() {
        this.message = "Hello, Vue.js!";
      },
      changeCount() {
        this.count++;
      },
    },
    beforeUpdate() {
      console.log("beforeUpdate called");
    },
  });
</script>
```

What will be logged to the console when the Change Count button is clicked?

Options :

6406532867199. ✖ 'beforeUpdate called'

6406532867200. ✔ Nothing will be logged on the console

6406532867201. ✖ An error will be thrown

6406532867202. ✖ 'beforeUpdate called' will be logged, followed by 'count updated'

Question Number : 119 Question Id : 640653852450 Question Type : MCQ
Correct Marks : 4.5

Question Label : Multiple Choice Question

Analyze the given HTML with embedded script.

```
<body>
  <div id="status"></div>
  <div id="result"></div>
  <script>
    function updateStatus(message) {
      document.getElementById("status").textContent = message;
    }
    function updateResult(message) {
      document.getElementById("result").textContent = message;
    }
    let promise1 = new Promise((resolve, reject) => {
      setTimeout(() => {
        resolve("Data Loaded");
      }, 1000);
    });

    let promise2 = Promise.reject("Network Error");
    promise1
      .then((result) => {
        updateStatus(result);
      })
      .catch((error) => {
        updateResult(error);
      });
    promise2
      .catch((error) => {
        updateStatus(error);
        return Promise.resolve("Retry Successful");
      })
      .then((result) => {
        updateResult(result);
      });
  </script>
</body>
```

What will be the final state of HTML elements, after all the setTimeout() have finished executing.

Options :

6406532867203. ✖ #status will be "Data Loaded", and #result will be "Data Save Failed".

6406532867204. ✖ #status will be "Network Error", and #result will be "Retry Successful".

6406532867205. ✔ #status will be "Data Loaded", and #result will be "Retry Successful".

6406532867206. ✖ `#status` will be "Network Error", and `#result` will be "Data Save Failed".

Question Number : 120 Question Id : 640653852453 Question Type : MCQ
Correct Marks : 4.5
Question Label : Multiple Choice Question

Consider the following HTML document embedded with Vue CDN and the script file shown below.

Filename: index.html

```
<div id="app">
    <main-comp>
        <nest-comp></nest-comp>
    </main-comp>
</div>
```

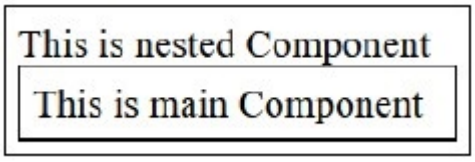
Filename: script.js

```
let main_comp = {
    template: `<div style="border: 1px solid black; padding: 5px">This is
main Component</div>`
}

let nest_comp = {
    template: `<div style="border: 1px solid black; padding: 5px">This is
nested Component</div>`
}

const app = new Vue({
    components: {
        "main-comp": main_comp,
        "nest-comp": nest_comp,
    }
}).$mount('#app')
```

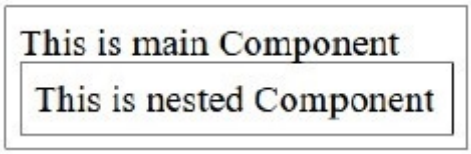
What will be rendered on the browser?

Options :

6406532867215. ✖



6406532867216. ✖



6406532867217. ✔ This is main Component

6406532867218. ✖ This is nested Component

Sub-Section Number : 6
Sub-Section Id : 640653126864
Question Shuffling Allowed : Yes

Question Number : 121 Question Id : 640653852455 Question Type : MSQ
Correct Marks : 4.5 Max. Selectable Options : 0
Question Label : Multiple Select Question

Consider the below Vue component template and script definitions that use Vue router.

Vue component template

```
<template>
  <div>
    <h1>{{ pageAction }}</h1>
    <router-link to="/login" v-if="showLogin">Login</router-link>
    <router-link to="/register" v-if="showRegister">Register</router-link>
    <router-view></router-view>
  </div>
</template>
```

Filename: script.js

```
<script>
export default {
  name: 'App',
  data() {
    return {
      pageAction: 'Login or Signup',
      showLogin: true,
      showRegister: false,
    };
  },
  watch: {
    '$route.path'() {
      if (this.$route.path === '/login') {
        this.showRegister = true;
        this.showLogin = false;
        this.pageAction = 'Login Page';
      } else if (this.$route.path === '/register') {
        this.showRegister = false;
        this.showLogin = true;
        this.pageAction = 'Register Page';
      } else {
        this.pageAction = 'Login or Signup';
      }
    },
  },
};
</script>
```

Assuming that the corresponding routes are properly configured, what does this component structure accomplish?

Options :

6406532867223. ✖ As soon as the app is run by a client, it displays a web page with a heading title "Login or Signup" and two navigation links to "Login" and "Register"

6406532867224. ✔ It dynamically updates the heading title based on the route and conditionally shows navigation links to "Login" and "Register"

6406532867225. ✖ It has a bug and will throw a runtime error.

6406532867226. ✔ The heading title is populated with the value “Login or Signup” when the user opens the app for the first time.

MLP

Section Id :	64065360930
Section Number :	8
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
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653126865
Question Shuffling Allowed :	No

Question Number : 122 Question Id : 640653852456 Question Type : MCQ

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING PRACTICE (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 :

6406532867227. ✔ YES

6406532867228. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653126866
Question Shuffling Allowed :	Yes

Question Number : 123 Question Id : 640653852457 Question Type : MCQ

Correct Marks : 3