

```

        if(obj instanceof ComplexNumber) {
            ComplexNumber c = obj;
            if(this.real == c.real && this.img == c.img)
                return true;
        }
6406532239940. ✖ return false;

        if(obj instanceof ComplexNumber) {
            ComplexNumber c = (ComplexNumber) obj;
            if(this.real == c.real && this.img == c.img)
                return true;
        }
6406532239941. ✔ return false;

```

AppDev2

Section Id :	64065344904
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
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 :	64065395184
Question Shuffling Allowed :	No

Is Section Default? : null

Question Number : 150 Question Id : 640653668591 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 :

6406532239978. ✓ YES

6406532239979. ✗ NO

Sub-Section Number : 2

Sub-Section Id : 64065395185

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 151 Question Id : 640653668592 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 regarding, var, let & const?

Options :

6406532239980.

✖ A variable created using "var" is always added as a property to the global object.

6406532239981. ✔ A variable created using "let" is never added as a property to the global object.

6406532239982. ✖ A variable created using "const" is not hoisted.

6406532239983. ✖ All of these

Question Number : 152 Question Id : 640653668594 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 not a valid IIFE (Immediately Invoked Function Expression)?

Options :

```
(function ( ) {  
    // some code here  
})();
```

6406532239988. ✖

```
( ( ) => {  
    // some code here  
})();
```

6406532239989. ✖

```
function ( ) {  
    // some code here  
} ( );
```

6406532239990. ✔

```
(function myIIFE (arg1, arg2) {  
    // some code here  
})(10, 20);
```

6406532239991. ✖

Question Number : 153 Question Id : 640653668598 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 not an example of application or UI (ephemeral) state?

Options :

6406532240004. ✖ User Preferences & Recommendations

6406532240005. ✖ Loading Icons

6406532240006. ✖ Dashboard Displays

6406532240007. ✔ User Database of NPTEL

Question Number : 154 Question Id : 640653668600 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 are correct regarding a variable declared using var keyword?

Options :

6406532240013. ✖ Variable is always globally scoped.

6406532240014. ✖ Variable declared inside the function can be accessed outside the function.

6406532240015. ✔ Variable declared inside a function can only be accessed inside the function.

6406532240016. ✖ None of these

Sub-Section Number : 3

Sub-Section Id : 64065395186

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 155 Question Id : 640653668593 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 correct definition(s) of an arrow function that accepts 2 numbers and return the addition of their squares?

Options :

6406532239984. ✖ `let func = (num1, num2) => return num1*num1 + num2*num2;`

6406532239985. ✔ `let func = (num1, num2) => num1*num1 + num2*num2;`

6406532239986. ✔ `let func = (num1, num2) => { return num1*num1 + num2*num2; }`

6406532239987. ✖ `let func = (num1, num2) => { num1*num1 + num2*num2; }`

Question Number : 156 Question Id : 640653668595 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 event loop in javascript?

Options :

6406532239992. ✔ The event loop makes sure that each function is processed (or executed) completely before any other function is executed.

6406532239993. ✔ The event loop takes the function from the callback queue, and sends the function to the call stack for execution.

6406532239994. ✖ The event loop can stop any function in running state to allow some other function to run on priority.

6406532239995. ✖ All of these

Sub-Section Number : 4
Sub-Section Id : 64065395187
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 157 Question Id : 640653668597 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

Which of the following shows the correct output for the below javascript program?

```
var a = 100;

const obj1 = {
  a : 10,
  b : 20,

  func : function (a) {
    console.log("Value:", a);
  }
}

const obj2 = {
  a : 40,
  b : 50,

  func : function (a) {
    this.func(a);
  }
}

obj2.func.apply(obj2, []);
```

Options :

6406532240000. ✖ Value: 100

6406532240001. ✔ The program will crash due to repeated function calls

6406532240002. ✖ Value: 40

6406532240003. ✖ Value: 10

Question Number : 158 Question Id : 640653668602 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.

```
const userDetails = {
  name: 'user1',
  email: 'user1@email.com',
  password: '12345',
}
function user({ userName: name = 'Name1', password }) {
  return `Username: ${name}, Password: ${password}`
}
console.log(user(userDetails))
```

What will be logged on to the console?

Options :

6406532240021. ✔ Username: Name1, Password: 12345

6406532240022. ✖ Username: user1, Password: 12345

6406532240023. ✖ Username: Name1, Password: undefined

6406532240024. ✖ Username: user1, Password: undefined

Question Number : 159 Question Id : 640653668604 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 code.

```
class Ball {
  constructor(batsman, run = 0, extra = false) {
    this.batsman = batsman
    this.run = run
    this.extra = extra
  }
}

class Over {
  constructor(overNo) {
    this.overNo = overNo
    this.balls = []
  }
}

class Inning {
  constructor(team, striker, nonStriker) {
    this.team = team
    this.striker = striker
    this.nonStriker = nonStriker
    this.overs = []
    this.currentOver = new Over(1)
  }
  changeStrike() {
    ;[this.striker, this.nonStriker] = [this.nonStriker,
this.striker]
  }
  changeOver() {
    this.currentOver = new Over(this.currentOver.overNo + 1)
    this.changeStrike()
  }
}

const inning = new Inning('India', 'Rohit', 'Gill')
const runs = [0, 4, 3, 6, 2, 6, 4, 1, 3]
runs.forEach((run, index) => {
  const ball = new Ball(inning.striker, run)
  inning.currentOver.balls.push(ball)
  if (run % 2 == 0) {
    inning.changeStrike()
  }
  if ((index + 1) % 3 == 0) {
    inning.overs.push(inning.currentOver)
    inning.changeOver()
  }
})
```

What will be logged on to the console for the statement

"console.log(inning.overs[0].balls[2].batsman)"?

Options :

6406532240029. ✔ Rohit

6406532240030. ✖ Gill

6406532240031. ✖ Virat

6406532240032. ✖ None of these

Question Number : 160 Question Id : 640653668605 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 code.

```

class Ball {
  constructor(batsman, run = 0, extra = false) {
    this.batsman = batsman
    this.run = run
    this.extra = extra
  }
}

class Over {
  constructor(overNo) {
    this.overNo = overNo
    this.balls = []
  }
}

class Inning {
  constructor(team, striker, nonStriker) {
    this.team = team
    this.striker = striker
    this.nonStriker = nonStriker
    this.overs = []
    this.currentOver = new Over(1)
  }
  changeStrike() {
    ;[this.striker, this.nonStriker] = [this.nonStriker,
this.striker]
  }
  changeOver() {
    this.currentOver = new Over(this.currentOver.overNo + 1)
    this.changeStrike()
  }
}

const inning = new Inning('India', 'Rohit', 'Gill')
const runs = [0, 4, 3, 6, 2, 6, 4, 1, 3]
runs.forEach((run, index) => {
  const ball = new Ball(inning.striker, run)
  inning.currentOver.balls.push(ball)
  if (run % 2 == 0) {
    inning.changeStrike()
  }
  if ((index + 1) % 3 == 0) {
    inning.overs.push(inning.currentOver)
    inning.changeOver()
  }
})

```

What will be logged on to the console for the statement "console.log(inning.currentOver.overNo)"?

Options :

6406532240033. ✖ 1

6406532240034. ✖ 2

6406532240035.

✖ 3

6406532240036. ✔ 4

Sub-Section Number : 5
Sub-Section Id : 64065395188
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 161 Question Id : 640653668596 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 is not a correct way to destructure (or unpack) an array?

Options :

6406532239996. ✖ `const array = [1, 2, 3, 4, 5]`
`const [a, b, c] = array`

6406532239997. ✖ `const array = [1, 2, 3, 4, 5]`
`const [a, , c] = array`

6406532239998. ✔ `const array = [1, 2, 3, 4, 5]`
`const [a, ...remaining, c] = array`

6406532239999. ✖ `const array = [1, 2, 3, 4, 5]`
`const [a, b, ...remaining] = array`

Question Number : 162 Question Id : 640653668599 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">
  Some Data for App
  <new>Fallback Data</new>
</div>
<script src = "app.js"> </script>
```

app.js:

```
Vue.component('new', {
  template : `<div>
    <slot>Main Data</slot>
    <h1>End of Data</h1>
  </div>
`
});

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

What will be rendered on the browser screen (ignore the formatting and styling)?

Options :

Some Data for App
Fallback Data
End of Data

6406532240008. ✓

Some Data for App
Main Data
End of Data

6406532240009. ✗

Some Data for App
Fallback Data
Main Data
End of Data

6406532240010. ✗

Some Data for App
End of Data

6406532240011. ✖

6406532240012. ✖ A warning will be issue for using an undefined tag, i.e., new

Question Number : 163 Question Id : 640653668601 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 code.

```
let sayHello = 'Hello from outside'
var greet = 'Greetings from outside'

{
  let sayHello = 'Hello from inside'
  var greet = 'Greetings from inside'
}

console.log(sayHello)
console.log(greet)
```

What will be logged on the console?

Options :

Hello from outside
Greetings from outside

6406532240017. ✖

Hello from inside
Greetings from inside

6406532240018. ✖

6406532240019. ✓
Hello from outside
Greetings from inside

6406532240020. ✖
Hello from inside
Greetings from outside

Question Number : 164 Question Id : 640653668603 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 code.

```
function getName() {  
    return this.userName  
}  
  
const getEmail = () => this.email  
  
const User1 = { userName: 'Test User1', email: 'user1@email.com' }  
const User2 = { userName: 'Test User2', email: 'user2@email.com' }  
  
User1.getName = getName  
User2.getEmail = getEmail  
  
console.log(User1.getName())  
console.log(User2.getEmail())
```

What will be logged on to the console?

Options :

6406532240025. ✓
Test User1
Undefined

6406532240026. ✖ Test User1
user2@email.com

6406532240027. ✖ Test User1
Test User2

6406532240028. ✖ Test User2
user2@email.com

Question Number : 165 Question Id : 640653668606 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 app.js

index.html

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

app.js

```
new Vue({
  el: '#app',
  template: `<div> Total Count: {{count}}</div>`,
  data: {
    count: 0,
  },
  beforeCreate() {
    this.count += 1
  },
  created() {
    this.count += 1
  },
  beforeMount() {
    this.count += 1
  },
  mounted() {
    this.count += 1
  },
})
```

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

Options :

6406532240037. ✖ Total Count: 2

6406532240038. ✔ Total Count: 3

6406532240039. ✖ Total Count: 4

6406532240040. ✖ Total Count: 5

Question Number : 166 Question Id : 640653668607 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 app.js

index.html

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

app.js

```
const Counter = {
  template: `<div> <b>{{counter}}</b><button @click='changeStatus'>
Click Me </button></div>`,
  props: ['initial'],
  data() {
    return {
      wasClicked: false,
      counter: this.initial,
    }
  },
  methods: {
    changeStatus() {
      this.wasClicked = !this.wasClicked
    },
  },
  watch: {
    wasClicked(val) {
      if (val) {
        this.counter += 1
      }
    },
  },
}

new Vue({
  el: '#app',
  template: `<div><counter :initial=3 /></div>`,
  components: {
    counter: Counter,
  },
})
```

Suppose the application is running on "<http://127.0.0.1:8080>". What will be the value of data property "counter" of "Counter" component if the user clicks on click me button 5 times.

Options :

6406532240041. ✖ 0

6406532240042. ✖ 5

6406532240043. ✔

MLT

Section Id :	64065344905
Section Number :	11
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	11
Number of Questions to be attempted :	11
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 :	64065395189
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 167 Question Id : 640653668608 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 : MACHINE LEARNING