Number of printed pages: 3

Semester V (B. Tech)

Er. No. 20183058

Academic Year: 2022-23

Jaypee University of Engineering & Technology, Guna

T-3 (Odd Semester 2022)

21B14CI742 – Full Stack Development

Maximum Duration: 2 Hours

Maximum Marks: 35

Notes:

1. This question paper has 5 questions.
2. Write relevant answers only.
3. Do not write anything on question paper (Except Er. No.)

		Marks	CO No.
Q1.	Read the following statements in context of Full Stack Development, and mark each of them as 'True' or 'False' and then write a short commentary (about 150 words) about the topics touched in that statement.	**[10]**	**CO4**
T	**(a)** Margin clears an area outside the border in CSS box model.		
F	**(b)** An HTTP request containing a form-field 'Salary Amount' must be sent through 'GET' method.		
F	**(c)** Pseudo-class names are not case-sensitive.		
T	**(d)** and tells nothing about its content, neither to the user nor to the browser.		
T	**(e)** A callback is a function passed as an argument to another function.		

Q2.	Using the concept of "CSS box model", compute the width of a element styled through the following CSS code:
	div {
	width: 320px;
	padding: 10px;
	border: 5px solid gray;
	margin: 0;
	}
		[05]	**CO2**

Q3.	Investigate the following piece of JavaScript code and report your findings under purpose, working and output as headings. const add = (function () {	**[10]**	**CO3**

```
  let counter = 0;
  return function () {counter += 1; return counter}
})();

add ();
add ();
add ();
console.log (add ());
```

Q4. Suppose a drug agency wants you to write a program using JavaScript callbacks such that the transactions from Distributor completes first, transactions from Wholesaler completes second, and from Retailer completes at last. A transaction first from a Wholesaler and then from a Distributor is termed invalid and must not happen at the first place. Similarly, a transaction from a Retailer in any other place except the last is termed invalid, thus the order of the transactions is important. The drug agency also has its IT department and they've already written the following piece of JavaScript code shown as section A and Section B, for convenience. When the underwritten program runs, it doesn't execute the transactions in required order. You are required to make changes in section B only, without altering any code in section A, such that the whole program makes sense and produce the expected output. **[05]** **CO4**

Expected output: DISTRIBUTOR has done Transaction
WHOLESALER has done Transaction
RETAILER has done Transaction

```
----------Section A Starts----------
let err;
function transact (title, callback) {
  let rand=Math.floor(Math.random() * 100; //generating rand no
  if (title.toUpperCase( )!=title)    //checking if title is in
                                      // uppercase
    err=new Error ("Uppercase Error");
  setTimeout(( )=>{callback(err)}, rand);
}
----------Section A Ends----------

----------Section B Starts----------
transact ("DISTRIBUTOR", (err)=> {
  console.log ("DISTRIBUTOR has done Transaction");
});
```

```
transact ("WHOLESALER", (err)=> {
  console.log ("WHOLESALER has done Transaction");
});

transact ("RETAILER", (err)=> {
  console.log ("RETAILER has done Transaction");
});
```

----------Section B Ends----------

Q5. "Suppose you were promised an ice-cream by your mother if you would clean the kitchen for her. Either you would clean the kitchen and grabbed your ice-cream, or you wouldn't have cleaned the kitchen and didn't get your reward." Relate the aforementioned case with JavaScript Promises and program the above scenario. **[05]** **CO4**