

JavaScript Essentials and ES6

Unit: 4

Web Technology (ACSE0516))

Course Details
(B Tech 5th Sem)



Praveen Tomar
Assistant Professor
CSE-AI
NIET, Greater Noida



Table of Contents

1. Name of Subject with code, Course and Subject Teacher
2. Brief Introduction of Faculty member with Photograph
3. Evaluation Scheme
4. Subject Syllabus
5. Branch wise Applications
6. Course Objective (Point wise)
7. Course Outcomes (COs)
8. Program Outcomes only heading (POs)
9. COs and POs Mapping
10. Program Specific Outcomes (PSOs)

- 11.CO and PSOs Mapping
- 12.Program Educational Objectives (PEOs)
- 13.Result Analysis (Department Result, Subject Result and Individual Faculty Result)
- 14.End Semester Question Paper Templates (Offline Pattern/Online Pattern)
- 15.Perquisite/ Recap
- 16.Brief Introduction about the Subject with videos
- 17.Unit Content
- 18.Unit Objective
- 19.Topic Objective/Topic Outcome
- 20.Lecture related to topic
- 21.Daily Quiz
- 22.Weekly Assignment

23 Topic Links

24 MCQ (End of Unit)

25 Glossary Questions

26 Old Question Papers (Sessional + University)

27 Expected Questions

28 Recap of Unit

Brief Introduction of Faculty



Dedicated and innovative computer science professional with a strong research background and a deep passion for technology. I hold a Master's degree in Computer Science . Over the past 13 years, I have been actively involved in research projects focused on Artificial Intelligence, machine learning, IOT. My expertise lies in web development in the field of computer science. Throughout my academic journey, I have contributed to several published research papers and have presented my work at prestigious conferences. Proficient in programming languages like C,C++,Python and Java, I possess a problem-solving mindset and a knack for identifying creative solutions to complex challenges.

Research Publications: 06

Patent: 01

Evaluation Scheme

Sl. No.	Departmental Electives	Subject Codes	Subject Name	Bucket Name	Branch	Semester
1	Elective-I	ACSE0511	CRM Fundamentals	CRM-RPA	DS	5
2	Elective-II	ACSE0513	CRM Administration		DS	5
3	Elective-I	ACSAI0516	Predictive Analytics	Data Analytics	DS	5
4	Elective-II	ACSE0516	Web Technology		DS	5
5	Elective-I	ACSE0512	Python Web Development with Django	Full Stack Development	DS	5
6	Elective-II	ACSE0514	Design Patterns		DS	5
7	Elective-I	ACSAI0515	Mobile Application Development	Mobility Management	DS	5
8	Elective-II	ACSAI0521	Development in Swift Fundamentals		DS	5

Syllabus Unit -1

Introduction: Introduction to Web Technology, History of Web and Internet, Connecting to Internet, Introduction to Internet services and tools, Client-Server Computing, Protocols Governing Web, Basic principles involved in developing a web site, Planning process, Types of Websites, Web Standards and W3C recommendations.

Web Hosting: Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel, Using FTP Client, Maintaining a Website.

Syllabus Unit -2

HTML: What is HTML, DOM- Introduction to Document Object Model, Basic structure of an HTML document, Mark up Tags, Heading-Paragraphs , Line Breaks, Understand the structure of HTML tables. Lists, Working with Hyperlinks, Image Handling, Understanding Frames and their needs, HTML forms for User inputs. New form Elements- date, number, range, email, search and data list, Understanding audio, video and article tags.

XML: Introduction, Tree, Syntax, Elements, Attributes, Namespaces, Display, HTTP request, Parser, DOM, XPath, XSLT, XQuery, XLink, Validator, DTD, Schema, Server

Syllabus Unit -3

Concept of CSS 3: Creating Style Sheet, CSS Properties, CSS Styling(Background, Text Format, Controlling Fonts) , Working with block elements and objects , Working with Lists and Tables , CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties) CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute selector) , CSS Color, Creating page Layout and Site.

Bootstrap: Introduction, Bootstrap grid system, Bootstrap Components.

Syllabus Unit -4

JavaScript Essentials: Introduction to Java Script , Java script Types , Var, Let and Const Keywords, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects ,JS Functions , Using Java Script in Real time , Validation of Forms, Arrow functions and default arguments, Template Strings, Strings methods, Callback functions,

Object de-structuring, Spread and Rest Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator,

Difference == & ===, Asynchronous Programming in ES6, Promise Constructor, Promise with Chain, Promise Race.

Syllabus Unit -5

Introduction to PHP, Basic Syntax, Variables & Constants, Data Type, Operator & Expressions, Control flow and Decision making statements, Functions, Strings, Arrays,

Working with files and directories: Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.

Session & Cookies: Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session.

Text books:

1. Steven M. Schafer, “HTML, XHTML, and CSS Bible, 5ed”, Wiley India
2. Ian Pouncey, Richard York, “Beginning CSS: Cascading Style Sheets for Web Design”, Wiley India

Sample Applications

- Desktop GUI Applications
- Mobile Applications
- Enterprise Applications
- Scientific Applications
- Web-based Applications
- Cloud-based Applications
- Web servers and Application servers
- Software Tools

Course Objective

- This course covers different aspect of web technology such as HTML, CSS, and issues of web technology, client and server side issue.
- The general objectives of this course are to **provide** fundamental concepts of Internet; Web Technology and Web Programming.
- Students will be able to build a proper responsive website.

Course Outcome

At the end of the semester, student will be able to:

Course Outcome s (CO)	CO Description	Blooms' Taxonomy
CO1	Recalling the basic facts and explaining the basic ideas of Web technology and web hosting.	K2
CO2	Applying and creating various HTML5 semantic elements and application with working on HTML forms for user input.	K6
CO3	Understanding and applying the concepts of Creating Style Sheet (CSS)3 and bootstrap.	K3
CO4	Analysing and implementing concept of Java Script and its applications.	K5
CO5	Creating and evaluating dynamic web pages using the concept of PHP	K6

- 1. Engineering knowledge:
- 2. Problem analysis:
- 3. Design/development of solutions:
- 4. Conduct investigations of complex problems:
- 5. Modern tool usage:
- 6. The engineer and society:
- 7. Environment and sustainability:
- 8. Ethics:
- 9. Individual and team work:
- 10. Communication:
- 11. Project management and finance:
- 12. Life-long learning

CO-PO Mapping

Mapping of Course Outcomes and Program Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ACSE0516 1.1	3	2	2	1	1	1	1	1	3	3	2	3
ACSE0516 1.2	3	2	3	1	3	2	2	1	3	3	3	3
ACSE0516 1.3	3	2	3	2	3	2	2	2	3	3	2	3
ACSE0516 1.4	3	3	3	2	3	2	2	2	3	3	2	3
ACSE0516 1.5	3	3	3	2	3	2	2	2	3	3	2	3
Average	3	2.4	2.8	1.6	2.6	1.8	1.8	1.6	3	3	2.2	3

Program Specific Outcomes

- **PSO1:** Work as a software developer, database administrator, tester or networking engineer for providing solutions to the real world and industrial problems.
- **PSO2:** Apply core subjects of information technology related to data structure and algorithm, software engineering, web technology, operating system, database and networking to solve complex IT problems.
- **PSO3:** Practice multi-disciplinary and modern computing techniques by lifelong learning to establish innovative career.
- **PSO4:** Work in a team or individual to manage projects with ethical concern to be a successful employee or employer in IT industry.

COs and PSOs Mapping

Mapping of Program Specific Outcomes and Course Outcomes:

Course Outcomes	Program Specific Outcomes			
	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2	3
CO2	3	3	3	2
CO3	3	3	2	2
CO4	3	2	3	2
CO5	3	2	2	2
AVG	3	2.6	2.4	2.2

- **PEO1:** able to apply sound knowledge in the field of information technology to fulfill the needs of IT industry.
- **PEO2:**able to design innovative and interdisciplinary systems through latest digital technologies.
- **PEO3:** able to inculcate professional and social ethics, team work and leadership for serving the society.
- **PEO4:** able to inculcate lifelong learning in the field of computing for successful career in organizations and R&D sectors.

End Semester Question Paper Template

B TECH

(SEM-V) THEORY EXAMINATION 20__-20__

Time: 3 Hours

Total

Marks: 100

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt all questions in brief.

1 x 10 =

Q.No.	Question	Marks	CO
1		1	
2		1	
.		.	
10		1	

End Semester Question Paper Templates

2. Attempt of the following:

5 x 2 = 20

Q.No.	Question	Marks	CO
1		2	
2		2	
“””		“”	
5		2	

SECTION B

3. Attempt any five part of the following:

5 x 6 = 30

Q.No.	Question	Marks	CO
1		6	
.		6	
7		6	

End Semester Question Paper Templates

4. Attempt any one part of the following:

1 x 10 = 10

Q.No.	Question	Marks	CO
1		10	
2		10	

5. Attempt any one part of the following:

1 x 10 = 10

Q.No.	Question	Marks	CO
1		10	
2		10	

6. Attempt any one part of the following:

1 x 10 = 10

Q.No.	Question	Marks	CO
1		10	
2		10	

End Semester Question Paper Templates

7. Attempt any one part of the following:

1 x 10 = 10

Q.No.	Question	Marks	CO
1		10	
2		10	

8. Attempt any one part of the following:

1 x 10 = 10

Q.No.	Question	Marks	CO
1		10	
2		10	

- Basic computer literacy, a basic understanding of HTML and CSS, familiarity with JavaScript
- basicsBasic Knowledge of any programming language like C/C++/Python/Java.
- Familiarity with basic concepts of Internet.

Brief Introduction About The Subject

- Web technologies are the various tools and techniques that are utilized in the process of communication between different types of devices over the internet.
- To understand this term in a better manner, let's break it down into two pieces: 'web' and 'technology'.
- The web, in this case, refers to the World Wide Web, more commonly known as WWW.
- It first came into being in 1989 when famous scientist and engineer, Tim Berners-Lee, came up with an efficient mechanism to share resources between scientists all over the world.
- https://www.youtube.com/results?search_query=Web+Technonogies

- Introduction to Java Script ,
- Java script Types , Var, Let and Const Keywords,
- Operators in JS ,
- Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events ,
- JS Arrays, Working with Arrays, JS Objects ,JS Functions ,
- Using Java Script in Real time , Validation of Forms, Arrow functions and default arguments,
- Template Strings, Strings methods, Callback functions,
- Object de-structuring, Spread and Rest Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator,
- Difference == & ===, Asynchronous Programming in ES6, Promise Constructor, Promise with Chain, Promise Race.

Unit 4 Objective

Objective of Unit 4:

- To understand the basics of working with objects in JavaScript: creating objects, accessing and modifying object properties using constructors. understating various function, loop, array, string etc.



Topic Objective/Outcome

Topics: Introduction to Java Script , Java script Types , Var, Let and Const Keywords, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects ,JS Functions , Using Java Script in Real time , Validation of Forms, Arrow functions and default arguments, Template Strings, Strings methods, Call back functions. **(CO4)**

Objective of the above topics:

- To know the basics of difference between HTML,CSS and JavaScript
- To get the knowledge about JAVA & JavaScript
- To learn how various datatypes, variables, keywords, operators etc.
- To learn various conditional statement, loop, types of function , how argument pass between function

Introduction to Java Script

- JavaScript is a lightweight, interpreted programming language.
- It is designed for creating network-centric applications
- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.
- JavaScript is easy to learn.

Java Script cont...

- Java Script is used insert dynamic text into HTML (ex: user name)
- It's get information about a user's computer (ex: browser type)
- It perform calculations on user's computer (ex: form validation)
- Java script offers a vast standard library that has a wide variety of functions and methods available to help in the process of development, making the entire process easier and hassle-free

Features of Java script

- Javascript is a lightweight, cross-platform, and interpreted language.
- It was developed with the main intention of DOM manipulation, bringing forth the era of dynamic websites.
- Javascript functions are objects and can be passed in other functions as parameters.
- Can handle date and time manipulation.
- Can perform Form Validation.
- A compiler is not needed.



+



=



Self-Invoking Functions

- Function expressions can be made "self-invoking".
- A self-invoking expression is invoked (started) automatically, without being called.
- Function expressions will execute automatically if the expression is followed by ().
- You cannot self-invoke a function declaration.
- we have to add parentheses around the function to indicate that it is a function expression:

Example

```
(function () {  
  let x = "Hello!!"; // I will invoke myself  
})();
```

Variables:-

```
var name = expression;
```

JS

```
var clientName = "Connie Client";  
var age = 32;  
var weight = 127.4;
```

JS

- variables are declared with the var keyword (case sensitive)
- types are not specified, but JS does have types ("loosely typed")
 - Number, Boolean, String, Array, Object, Function, Null, Undefined
 - can find out a variable's type by calling typeof

Number type

```
var enrollment = 99;  
var medianGrade = 2.8;  
var credits = 5 + 4 + (2 * 3);
```

JS

- integers and real numbers are the same type (no int vs. double)
- same operators: + - * / % ++ -- = += -= *= /= %=
- similar precedence to Java
- many operators auto-convert types: "2" * 3 is 6

Comments (same as Java)

```
// single-line comment  
/* multi-line comment */
```

JS

- identical to Java's comment syntax
- recall: 4 comment syntaxes
 - HTML: `<!-- comment -->`
 - CSS/JS/PHP: `/* comment */`
 - Java/JS/PHP: `// comment`
 - PHP: `# comment`

Math object

```
var rand1to10 = Math.floor(Math.random() * 10 + 1);  
var three = Math.floor(Math.PI);
```

JS

- **methods:** abs, ceil, cos, floor, log, max, min, pow, random, round, sin, sqrt, tan
- **properties:** E, PI

Logical operators

- `> < >= <= && || ! == != === !==`
- most logical operators automatically convert types:
 - `5 < "7"` is true
 - `42 == 42.0` is true
 - `"5.0" == 5` is true
- `===` and `!==` are strict equality tests; checks both type and value
- `"5.0" === 5` is false

if/else statement (same as Java)

```
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else {  
    statements;  
}
```

JS

- identical structure to Java's if/else statement
- JavaScript allows almost anything as a condition

Boolean type

```
var iLike190M = true;  
var ieIsGood = "IE6" > 0; // false  
if ("web devevelopment is great") { /* true */ }  
if (0) { /* false */ }
```

JS

- any value can be used as a Boolean
 - "falsey" values: 0, 0.0, NaN, "", null, and undefined
 - "truthy" values: anything else
- converting a value into a Boolean explicitly:
 - `var boolValue = Boolean(otherValue);`
 - `var boolValue = !! (otherValue);`

for loop (same as Java)

```
var sum = 0;  
for (var i = 0; i < 100; i++) {  
    sum = sum + i;  
}  
JS
```

```
var s1 = "hello";  
var s2 = "";  
for (var i = 0; i < s.length; i++) {  
    s2 += s1.charAt(i) + s1.charAt(i);  
}  
// s2 stores "hheelllloo"  
JS
```

while loops (same as Java)

```
while (condition) {  
    statements;  
}
```

JS

```
do {  
    statements;  
} while (condition);
```

JS

break and continue keywords also behave as in Java

Q1. Java script is an _____ language?

- (A) object oriented
- (b) object base
- © procedural
- (d) None of these

Q2. Which of the following keywords is used to define a variable in JavaScript?

- (a) let
- (b) var
- (C) both A & B
- (d) None of the above

Q3. How can a datatype be declared to be constant type ?

- (A) var
- (b) const
- © let
- (D) constant

Q4.Java script is ideal to

- (A)make computations in HTML simpler
- (B) minimize storage requirements on the web server
- (c) increase the download time for the client
- (D) none of the mentioned

Q5.The basic difference between javascript & java is

- (A)There is no difference
- (B). Functions are considered as fields
- ©. Variables are specific
- (D). Functions are values, and there is no hard distinction between methods and fields

Q6. Which of the following is not javascript framework?

- (A) Node
- (B) Vue
- (C) React
- (D) Cassandra

Q. 7 The three important manipulations done in a for loop on a loop variable?

- a) Updation, Incrementation, Initialization
- b) Initialization, Testing, Updation
- c) Testing, Updation, Testing
- d) Initialization, Testing, Incrementation

Q. 8 One of the special features of an interpreter in reference with the for loop is that

- a) Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property
- b) The iterations can be infinite when an interpreter is used
- c) The body of the loop is executed only once
- d) the iteration is finite when an interpreter is used

Q. 9 Among the keywords below, which one is not a statement?

- a) debugger
- b) with
- c) if
- d) use strict

Q 10 What will be the output of the following JavaScript code?

```
var a = 10;  
do {  
    a += 1;  
    console.log(a);  
} while (a < 5);
```

- a) 11121314
- b) 1112
- c) 12345
- d) 11

Q 11 What will be the output of the following JavaScript code?

```
var a = 0;  
var b = 0;  
while (a < 3)  
{  
    a++;  
    b += a;  
    console.log(b);  
}
```

- a) 135
- b) 123
- c) 013
- d) 01

Q 12 The expression `score = score + 10` can be written as

- `score += 10;`
- `score ++= 10;`
- `score =+ 10;`
- `score + 10 = 10;`

Q 13 The correct result of `score = 2 + 4 + "3"` is:

- 9
- 27
- 63
- 6

Q 14. Assignment Operators is following type of operator _____.

- None of these
- Unary
- Binary
- Ternary

Q 15 "Add and Assignment" operator is shown by this symbol.

`==+`

`+=`

`=+`

`+=`

Daily Quiz(Cont..)

Q 16. Among the following, which one is a ternary operator in JavaScript?

- A. # B. ::
C. &: D. ?:

Q 17. What does javascript use instead of == and !=?

- A. It uses bitwise checking B. It uses === and !== instead
C. It uses equals() and notequals() instead D. It uses equalto()

Q 18. What is the output of following Javascript?

```
var a = 'letsfind';  
var b = 'course';  
var c = a/b;  
document.write(c);
```

- A. Letsfindcourse B. letsfind/course
C. NaN D. None of the above

Q 19. How ++ works in Javascript? Find output of below Javascript code.

- A. 00 B. 01
C. 11 D. 10

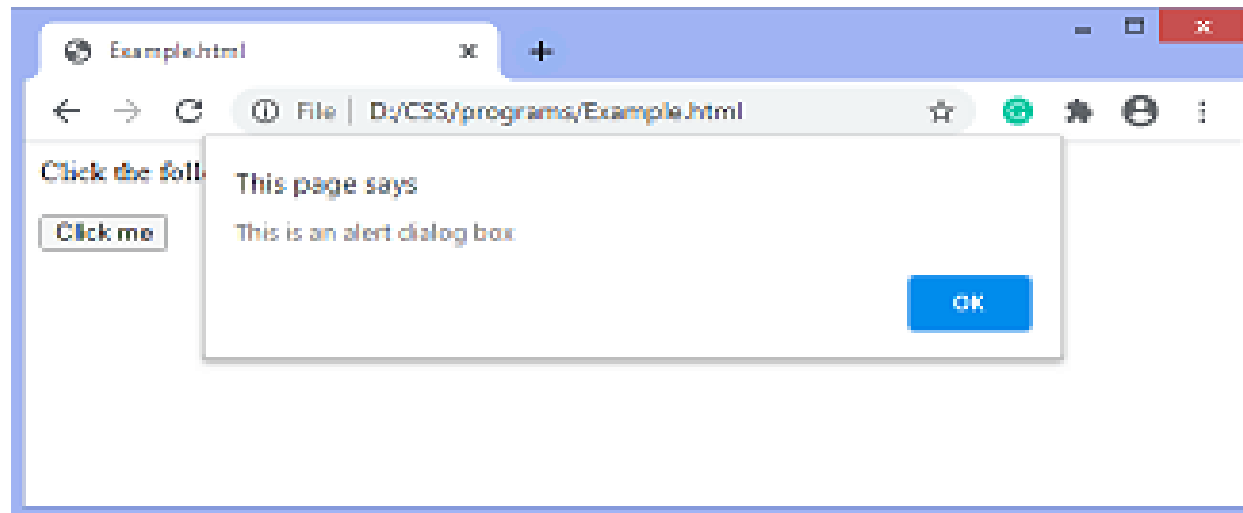
Daily Quiz (Cont..)

Q 20. Which of the following method checks if its argument is not a number?

- A. `isNaN()`
- B. `nonNaN()`
- C. `NaN()`
- D. None of the above

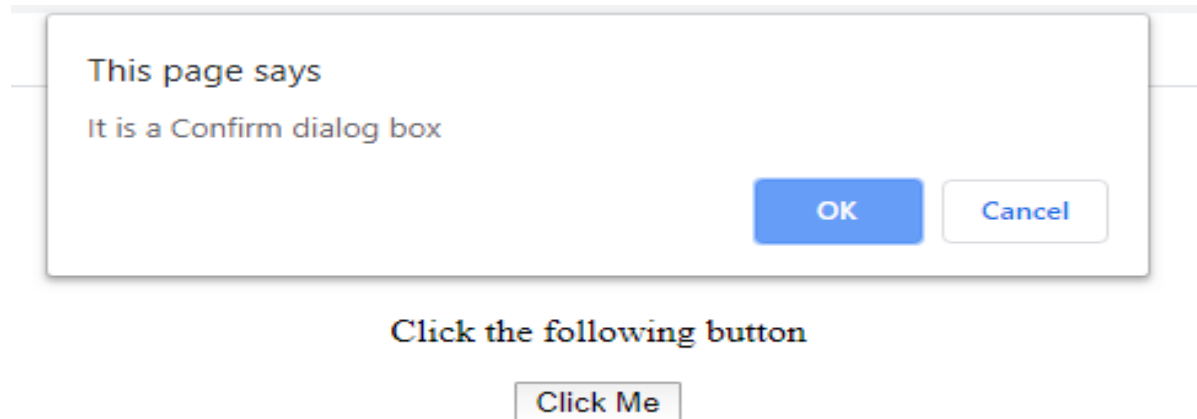
alert

- The **alert()** method in JavaScript is used to display a virtual alert box.
- It is mostly used to give a warning message to the users.



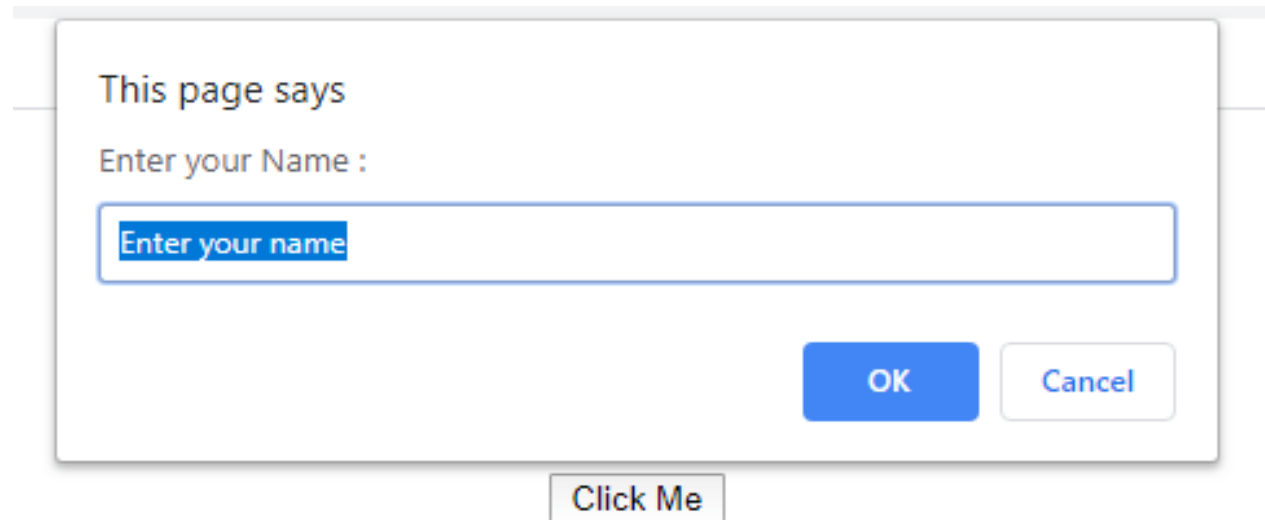
Confirmation

- It is widely used for taking the opinion from the user on the specific option.
- It includes two buttons, which are OK and Cancel. `



Prompt Dialog box

- The prompt dialog box is used when it is required to pop-up a text box for getting the user input.
- It enables interaction with the user.



Java script Events

- The change in the state of an object is known as an Event.
- represents that some activity is performed by the user or by the browser.
- When javascript code is included in HTML, js react over these events and allow the execution.
- Java Script handles the HTML events via Event Handlers.

Common HTML Events

Common HTML Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

JavaScript Event Handlers

- Things that should be done every time a page loads
- Things that should be done when the page is closed
- Action that should be performed when a user clicks a button
- Content that should be verified when a user inputs data
- HTML event attributes can execute JavaScript code directly
- HTML event attributes can call JavaScript functions
- You can assign your own event handler functions to HTML elements
- You can prevent events from being sent or being handled

JS Arrays

```
var name = []; // empty array  
var name = [value, value, ..., value]; // pre-filled  
name[index] = value; // store element
```

JS

```
var ducks = ["Huey", "Dewey", "Louie"];  
var stooges = []; // stooges.length is 0  
stooges[0] = "Larry"; // stooges.length is 1  
stooges[1] = "Moe"; // stooges.length is 2  
stooges[4] = "Curly"; // stooges.length is 5  
stooges[4] = "Shemp"; // stooges.length is 5
```

JS

Working with Arrays

```
var a = ["Stef", "Jason"]; // Stef, Jason  
a.push("Brian"); // Stef, Jason, Brian  
a.unshift("Kelly"); // Kelly, Stef, Jason, Brian  
a.pop(); // Kelly, Stef, Jason  
a.shift(); // Stef, Jason  
a.sort(); // Jason, Stef
```

JS

- push and pop add / remove from back
- unshift and shift add / remove from front
- shift and pop return the element that is removed

Q 1 What are the types of Pop up boxes available in JavaScript?

- a) Prompt
- b) Alert
- c) Confirm
- d) All of the above

Q 2 Which of the following is the correct syntax to display “Tutoriallinks” in an alert box using JavaScript?

- e) `alertbox("Tutoriallinks");`
- f) `message("Tutoriallinks");`
- g) `alert("Tutoriallinks");`
- h) All of the above

Q 3 Which one of the following JavaScript statements produces the output as shown?

- i) `alert("Please Enter Your Name:", "Your Name");`
- j) `confirm("Please Enter Your Name:", "Your Name");`
- k) `alertbox("Please Enter Your Name:", "Your Name");`
- l) `prompt("Please Enter Your Name:", "Your Name');`

Daily Quiz(cont..)

Q 4. The type that specifies what kind of event occurred is _____

- a) event type
- b) even target
- c) both event type and even target
- d) Interface

Q 5 Which is the object on which the event occurred or with which the event is associated?

- a) event type
- b) event target
- c) both event type and even target
- d) Interface

Q 6 In general, event handler is nothing but _____

- a) function
- b) interface
- c) event
- d) Handler

Q 7 The property specifies the property of the event-----

- a) Type
- b) Target
- c) Manner
- d) Program

Daily Quiz(cont..)

Q 8 The events that are directly tied to a specific input device are _____

- a) Device-independent input events
- b) Device-dependent input events
- c) User interface events
- d) State change events

Q 9 -----form of event propagation handles the registered container elements

- a) Event Propagation
- b) Event Registration
- c) Event Capturing
- d) Default Actions

Q 10 The high-level events among the following events are _____

- a) User interface events
- b) Device-independent events
- c) Device-dependent events
- d) Stage event change

Q 11 When will the browser invoke the handler?

- a) Program begins
- b) Any event occurs
- c) Specified event occurs
- d) Webpage loads

Q 12 In JavaScript, an array can be created using the:

- a. Array() constructor
- b. [] literals
- c. Both of the above

Q 13 How to access the second element of the array `arr = [3, 7, 10]`?

- a. `arr.indexOf(2)`
- b. `arr.indexOf(1)`
- c. `Carr[2]`
- d. `arr[1]`

Q 14 Arrays can have arrays as elements. True or false?

- a. True
- b. False

Q 15 Which property of array objects can be used to iterate over them?

- a. `length`
- b. `len`
- c. `Size`

Q 16 The method or operator used to identify the array is _____

- a) `isArrayType()`
- b) `==`
- c) `===`
- d) `typeof`

Q 17 The pop() method of the array does which of the following task?

- a) decrements the total length by 1
- b) increments the total length by 1
- c) prints the first element but no effect on the length
- d) updates the element

Q 18 What is the output?

```
const arr = ["Apple", "Banana", "Orange"];  
console.log(arr["2"]);
```

- a. undefined
- b. "Orange"
- c. TypeError: Invalid array index
- d. "Banana"

Q 19 Which method removes the last element from an array and returns that element?

- a. pop()
- b. shift()
- c. unshift()
- d. slice()


Q 20 Which method adds a new item to the end of an array?

- a. unshift()
- b. slice()
- c. push()
- d. pop()

JavaScript Objects

Real Life Objects, Properties, and Methods

In real life, a car is an **object**. A car has **properties** like weight and color, and **methods** like start and stop:

Object	Properties	Methods
	<code>car.name = Fiat</code>	<code>car.start()</code>
	<code>car.model = 500</code>	<code>car.drive()</code>
	<code>car.weight = 850kg</code>	<code>car.brake()</code>
	<code>car.color = white</code>	<code>car.stop()</code>

- All cars have the same properties, but the property values differ from car to car.
- All cars have the same methods, but the methods are performed at different times.

JavaScript Objects

JavaScript Objects

- This code assigns a **simple value** (Fiat) to a **variable** named car:

```
let car = "Fiat";
```

Objects are variables too. But objects can contain many values.

this code assigns **many values** (Fiat, 500, white) to a **variable** named car:

```
const car = {type:"Fiat", model:"500", color:"white"};
```

Object Definition

- You define (and create) a JavaScript object with an object literal:

Example

```
const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

- Spaces and line breaks are not important. An object definition can span multiple lines:

```
const person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50,  
  eyeColor: "blue"  
};
```

Object Properties

The **name : values** pairs in JavaScript objects are called **properties**

Property	Property Value
firstName	John
lastName	Doe
age	50
eyeColor	blue

Accessing Object Properties

- You can access object properties in two ways:
- *objectName.propertyName*
- Object Methods
- Objects can also have **methods**.
- Methods are **actions** that can be performed on objects.
- Methods are stored in properties as **function definitions**.

JavaScript Function Definitions

- Javascript functions are defined with **function keyword**
- You can use a function declaration or a function expression

Function Declarations

```
function functionName(parameters) {  
    // code to be executed  
}
```

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are invoked (called upon).

```
function myFunction(a, b) {  
    return a * b;  
}
```

Function Expressions

A JavaScript function can also be defined using an **expression**

A function expression can be stored in a variable:

```
const x = function (a, b) {return a * b};
```

- ❖ After a function expression has been stored in a variable, the variable can be used as a function:

```
const x = function (a, b) {return a * b};  
let z = x(4, 3);
```

- ❖ The function above is actually an **anonymous function** (a function without a name).
- ❖ Functions stored in variables do not need function names. They are always invoked (called) using the variable name.

The Function() Constructor

- ❖ Function can be defined with a built-in java script function constructor called function()

```
const myFunction = new Function("a", "b", "return a  
*b");
```

```
let x = myFunction(4, 3);
```

Self-Invoking Functions

- Function expressions can be made "self-invoking".
- A self-invoking expression is invoked (started) automatically, without being called.
- Function expressions will execute automatically if the expression is followed by ().
- You cannot self-invoke a function declaration.
- we have to add parentheses around the function to indicate that it is a function expression:

Example

```
(function () {  
    let x = "Hello!!"; // I will invoke myself  
})();
```

- The function above is actually an anonymous self-invoking function (function without name)

Arrow Functions

Arrow functions allows a short syntax for writing function expressions.

Example :

```
// ES5
var x = function(x, y) {
    return x * y;
}
```

```
// ES6
const x = (x, y) => x * y;
```

- ☐ Arrow functions do not have their own this. They are not well suited for defining object methods
- ☐ Arrow functions are not hoisted. They must be defined before they are used
- ☐ Using const is safer than using var, because a function expression is always constant value.
- ☐ You can only omit the return keyword and the curly brackets if the function is a single statement. Because of this, it might be a good habit to always keep them:

Example

```
const x = (x, y) => { return x * y };
```

JavaScript Function Parameters

A JavaScript **function** does not perform any checking on parameter values (arguments).

```
function functionName(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

Function parameters are the names listed in the function definition.

Function arguments are the real values passed to (and received by) the function

Parameter Rules

- JavaScript function definitions do not specify data types for parameters.
- JavaScript functions do not perform type checking on the passed arguments.
- JavaScript functions do not check the number of arguments received.

Default Parameters

- If a function is called with missing arguments (less than declared), the missing values are set to undefined

```
function myFunction(x, y) {  
    if (y === undefined) {  
        y = 2;  
    }  
}
```

Q 1 Which of the following is the literal way to create an object with a property x of value 10?

- A. {x: 10}
- B. new Object({x: 10})

Q 2 Suppose that obj is a pure JavaScript object. What will typeof obj return?

- A. "object"
- B. "obj"
- C. "pure object"

Q 3 Objects can contain objects as property values. True or false?

- A. True
- B. False

Q 4 A method of an object is simply a property with a function definition as its value. True or false?

- A. True
- B. False

Q 5 What is the constructor way to create an empty object?

- A. {}
- B. new Object()

Daily Quiz(Cont..)

Q 6 JavaScript is a pure object-based language. True or false?

- A. True
- B. False

Q 7 Javascript is an _____ language.

- A. object-oriented
- B. Procedural
- C. Object-based
- D. Prototype

Q 8 Object is the process of converting an objects state to a string from which it can later be restored.

- A. prototype
- B. class
- C. serialization
- D. Extensible

Q 9 The _____ operator is used to create an instance of an object.

- A. this
- B. self
- C. find
- D. New

Q 10 JavaScript provides a special constructor function called Object() to build the object.

- A. TRUE
- B. FALSE

Q 12 Do functions in JavaScript necessarily return a value?

- A. It is mandatory
- B. Not necessary
- C. Few functions return values by default
- D. some functions do not return any value

Q 13 What will be the output of the following JavaScript code?

```
var string2Num=parseInt("123xyz");
```

- A. 123
- B. 123xyz
- C. Exception
- D. NaN

Q 14 Which of the following function of Number object returns the number's value?

- A. toString()
- B. valueOf()
- C. toLocaleString()
- D. toPrecision()

Q 15 What surrounds a string?

- Quotations
- Curly Brackets
- Parenthesis
- Square Brackets

Daily Quiz(Cont..)

Q 16 functions need not allow invocations with zero arguments.

- A. zero
- B. strict
- C. empty
- D. Varargs

Q 17 A _____ is a group of reusable code which can be called anywhere in your program.

- A. exception
- B. function
- C. loop
- D. Switch

Q 18 The most common way to define a function in JavaScript is by using the _____ keyword.

- A. fun
- B. var
- C. function
- D. Define

Q 19 JavaScript allows us to write our own functions as well(True/False)

- A. True
- B. False

Q 20 The Function() constructor expects _____ number of string arguments

- A. 0
- B. 1
- C. 2
- D. any

Topic Objective/Outcome

Topics: Object de-structuring, Spread and Rest Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator, Difference == & ===, Asynchronous Programming in ES6, Promise Constructor, Promise with Chain, **.(CO4)**

Objective of the above topics:

- To learn the fundamentals of typescript
- To learn the creation of object in ES6.
- To get the knowledge about Programming in ES6.
- To learn the OOPS concepts by using ES6 Programming.
- To learn the concept of ES6 constructor chaning

TypeScript Fundamentals

- ❑ TypeScript is an **open-source language** that provides support for building enterprise scale JavaScript applications. Although several patterns exist that can be used to structure JavaScript,
- ❑ TypeScript is a programming language developed by Microsoft that is a superset of JavaScript. It is designed for developing enterprise scale JavaScript applications
- ❑ TypeScript provides **container functionality** that object-oriented developers are familiar with, such as classes and modules.
- ❑ It also supports strongly-typed code to ensure inappropriate values aren't assigned to variables in an application.

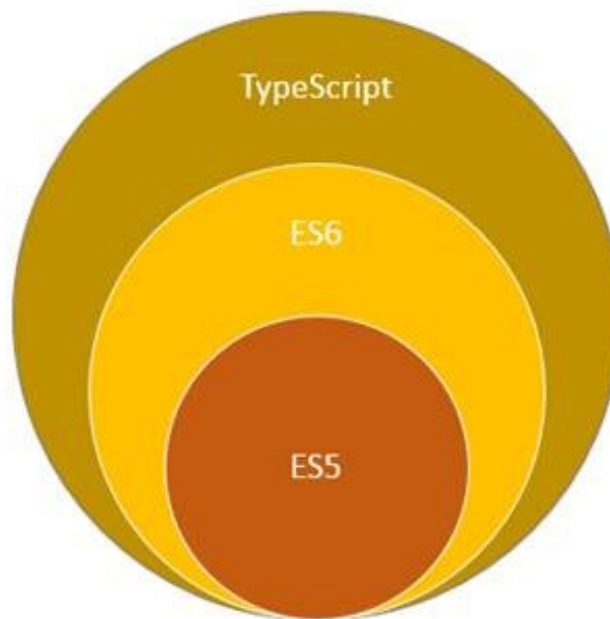
Difference between JavaScript & TypeScript

The main difference is that JavaScript is a scripting language used to make interactive web pages and applications, while **TypeScript is a superset of JavaScript**. Other differences include:

- ☐ TypeScript code needs to be compiled, but JavaScript does not
- ☐ TypeScript uses *types* and *interfaces* while JavaScript does not
- ☐ TypeScript has some additional features for functions
- ☐ TypeScript has a certain feature of prototyping that JS does not
- ☐ Maintaining a huge code in JavaScript is extremely difficult.

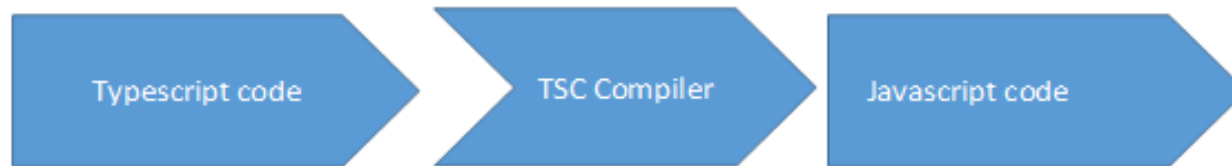
Key features of Typescript

- TypeScript is a free and open-source programming language developed and maintained by Microsoft.
- It is a superset of JavaScript, So you can write JavaScript code in typescript.
- Enables static typing and class-based object-oriented programming to JavaScript.
- Another key feature is that Angular 2 is written in typescript



How typescript works

- Typescript compiler converts code written in typescript to JavaScript,



Supported Editors We can write typescript using any one of the below editors.

- ✓ Visual Studio.
- ✓ Node.js
- ✓ Sublime Editor
- ✓ TypeScript play ground

Q 1 Which of the following best describes TypeScript?

- a. A programming language
- b. A markup language
- c. A scripting language
- d. A styling language

Q 2 What is the inherited type for the variable example in 'const example = ['Dylan']'?

- A) any[]
- B) string[]
- C) string
- D) unknown[]

Q 3 TypeScript will always correctly infer the type of an array.(True/False)

- A)True
- B)False

Q 4 .How many components typescript has?

- A)3
- B)4
- C)5
- D)6

Q 5 TypeScript supports how many types of comments

- A)1
- B)2
- C)3
- D)4

Q 6 TypeScript is a:

- A. strongly typed
- B. object oriented
- C. compiled language
- D. All of the above

Q 7 Which of the following are features of typeScript?

- A. TypeScript is just JavaScript
- B. TypeScript supports other JS libraries
- C. TypeScript is portable
- D. All of the above

Q 8 Extension of typescript is:

- A. .d.ty
- B. .d.tp
- C. .d.ts
- D. .d.td

Q 9 TypeScript supports Object Oriented Programming concepts like classes, interfaces, inheritance.

- A. TRUE
- B. FALSE

Q 10 Semicolons are optional in TypeScript.

- A. Yes
- B. No

Q 11 Which of the following companies has developed and designed TypeScript?

- a. Amazon
- b. TypeScript
- c. Microsoft
- d. Oracle

Q 12 What is the typing principle of typescript?

- a. Gradual
- b. Duck
- c. Dynamic
- d. All of the above

Q 13 Which of the computer programming languages below has influenced the creation of typescript?

- a. JavaScript
- b. Java
- c. C#
- d. All of the above

Q 14 What variable scopes are available in TypeScript?

- a. Global Scope
- b. Local Scope
- c. Class Scope
- d. All of the above

Q 15 What object oriented terms does Typescript support?

- a. Interfaces
- b. Classes
- c. Modules
- d. All of the above

Q 16. What are considered the types of access modifiers supported by TypeScript?

- a. Public
- b. Private
- c. Protected
- d. All of the above

Q 17 What does Typescript use for anonymous functions?

- e. "dot" syntax
- f. "arrow" syntax
- g. none of the above

Q 18 A typescript may be installed or managed through...

- h. Tag
- i. Nmp
- j. Space
- k. Void

Q 19 What are the benefits of Typescript?

- A. It helps in code structuring
- B. Use class based object oriented programming
- C. Impose coding guidelines
- D. All of the above

Q 20 TypeScript is :

- A. case-sensitive
- B. Case-insensitive
- C. depends on typescript version
- D. Can not say

Type Script Interfaces

- Interface is a structure that defines the contract in your application.
- It defines the syntax for classes to follow.
- Classes that are derived from an interface must follow the structure provided by their interface.
- The TypeScript compiler does not convert interface to JavaScript.
- It uses interface for type checking. This is also known as "duck typing" or "structural subtyping“
- An interface is defined with the keyword interface and it can include properties and method declarations using a function or an arrow function.

Example: Interface

```
interface IEmployee {  
    empCode: number;  
    empName: string;  
    getSalary: (number) => number; // arrow function  
    getManagerName(number): string;  
}
```

Interface as Type

- Interface in TypeScript can be used to define a type and also to implement it in the class.
- The following interface IEmployee defines a type of a variable.

Example: Interface as Type Copy

```
interface KeyPair {  
    key: number;  
    value: string;  
}
```

```
let kv1: KeyPair = { key:1, value:"Steve" }; // OK
```

```
let kv2: KeyPair = { key:1, val:"Steve" }; // Compiler Error: 'val' doesn't exist in type  
'KeyPair'
```

```
let kv3: KeyPair = { key:1, value:100 }; // Compiler Error:
```

Interface as Function Type

- TypeScript interface is also used to define a type of a function.
- This ensures the function signature.

Example: Function Type Copy

```
interface KeyValueProcessor
```

```
{  
    (key: number, value: string): void;  
};
```

```
function addKeyValue(key:number, value:string):void {  
    console.log('addKeyValue: key = ' + key + ', value = ' + value)  
}  
  
function updateKeyValue(key: number, value:string):void {  
    console.log('updateKeyValue: key = '+ key + ', value = ' + value)  
}
```


Interface as Function Type

- TypeScript interface is also used to define a type of a function.
- This ensures the function signature.

Example: Function Type Copy

```
interface KeyValueProcessor
```

```
{  
    (key: number, value: string): void;  
};
```

```
function addKeyValue(key:number, value:string):void {  
    console.log('addKeyValue: key = ' + key + ', value = ' + value)  
}  
  
function updateKeyValue(key: number, value:string):void {  
    console.log('updateKeyValue: key = '+ key + ', value = ' + value)  
}
```

Constructor in TypeScript

A constructor is a special function of the class that is automatically invoked when we create an instance of the class in Typescript.

We use it to initialize the properties of the current instance of the class.

Using the constructor parameter properties or Parameter shorthand syntax, we can add new properties to the class.

We can also create multiple constructors using the technique of constructor method overload.

Creating a Class Constructor

- The constructor method in a class must have the name constructor.
- A class can have only one implementation of the constructor method.
- The constructor method is invoked every time we create an instance from the class using the new operator.
- It always returns the newly created object.

Example of Constructor

```
class Person {
```

```
    constructor() {  
        console.log("Constructor is called")  
    }  
}
```

```
let p1= new Person()    //constructor is called
```

```
let p2= new Person()    //constructor is called
```

Parameters to the Constructor method

Constructors are just like normal functions in the way that they also accept parameters.

We need to pass parameter values when we create a new class instance.

Example:

```
class Person {  
    firstName:string;  
    lastName:string;  
  
    constructor(fName:string, lName:string) {  
        this.firstName=fName;  
        this.lastName=lName;  
    }  
}
```

Q 1 An Interface is a structure which acts as a contract in our application(True/False)

- A. True
- B. False

Q 2 The TypeScript compiler uses interface for type-checking (also known as "duck typing" or "structural subtyping")(True/False)

- C. True
- D. False

Q 3 The interface contains only the declaration of the methods and fields, but also the implementation(True/False)

- E. True
- F. False

Q 4 An interface is a keyword which is used to declare a TypeScript Interface.(True/False)

- G. True
- H. B. False

Q 5 We can inherit the interface from the other interfaces by using 'extends' keyword(True/False)

- I. True
- J. B. False

Q 6 We can use interfaces to describe the array type(Right/Wrong)

- K. Right
- L. Wrong

Q 7 Interface defines the required functions, and the class is responsible for implementing it to meet that contract(True/False)

- M. True
- N. False

Daily Quiz(Cont..)

Q 8 An interface type objects cannot declare any new methods or variables.(T/F)

- A. True.
- B. False

Q 9 In Typescript Interface are classes that contain body-less structure (abstract or virtual functions) (True/False)

- C. True
- D. False

Q 10 Interface enforces the variables and methods which have not to be present in an object. (True/False)

- E. True
- F. False

Q 11 TypeScript also allows us to use the interface in a class(True/False)

- G. True
- H. B. False

Q 12 When the Typescript compiler compiles it into JavaScript, then the interface will be disappeared from the JavaScript file(True/False)

- I. True
- J. False

Q 13 class Bird {

Bird () {console.log("I am Bird");}}

What will be output of : var obj=new Bird();?

- a.Error
- b.I am Birdc.
- c. Bird am I
- d. None of these

Q 14 _____ an interface will have the same properties as that interface.

- A)Idolizing
- B)Improving
- C)Extending
- D)Duplicating

Q 15 The constructor method is invoked every time we create an instance from the class using the new operator.
(True/False)

- A. True
- B. False

Q 16 Constructor need to pass parameter values when we create a new class instance.(True/False)

- A. True
- B. False

Q 18 The constructor method in a class must have the name constructor and not always returns the newly created object(True/False)

- C. True
- D. False

Daily Quiz(Cont..)

Q 19 How can you call base class constructor in Typescript?

- A. By using base()
- B. By using super()

Q 20 With a constructor on the interface, you can specify that all of your types must have certain methods/properties(True/False)

- C. True
- D. False

Q 21 TypeScript interface is not used to define a type of a function(True/False)

- E. True
- F. B. False

Q 22 Can you have multiple constructor implementation in Typescript?(Yes/No)

- G. Yes.
- H. No.

Q 23 Constructors are identified with the keyword "constructor" in Typescript(Right/Wrong)

- I. Right
- J. Wrong

Q 24 A constructor is a special function of the class that is responsible for initializing the variables of the class(True/False)

- K. True
- L. B. False

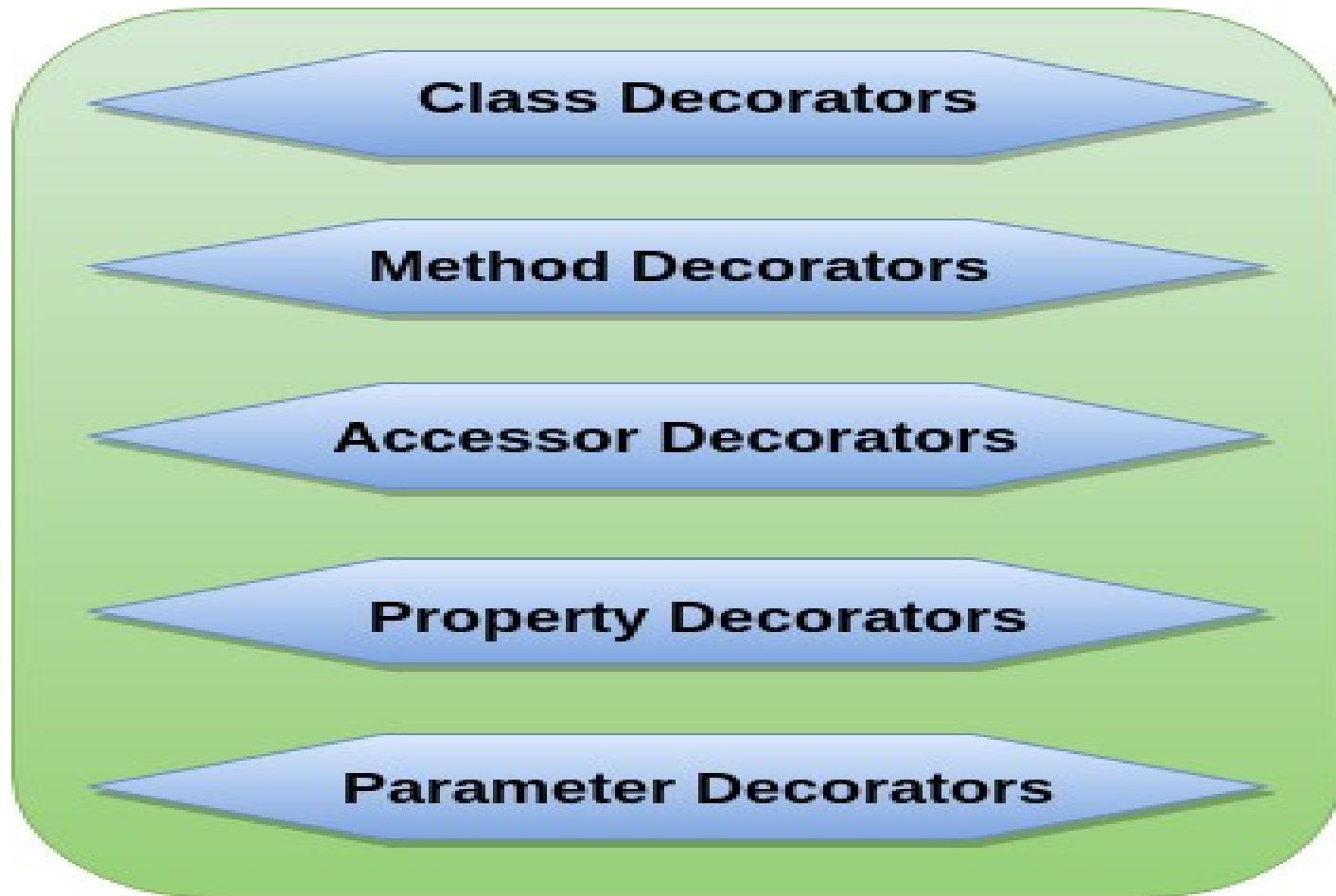
Q 25 Constructor is special type of function that always has a return type(Right/Wrong)

- M. Right
- N. B. Wrong

TypeScript Decorators

- A Decorator is a special kind of declaration that can be applied to classes, methods, accessor, property, or parameter.
- Decorators are simply functions that are prefixed **@expression** symbol, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration.
- Decorators are an experimental feature proposed for ES7.
- It is already in use by some of the JavaScript frameworks including Angular 2.
- TypeScript Decorators serves the purpose of adding both annotations and metadata to the existing code in a declarative way.

Types of Decorators



Types of Decorators

Class Decorators

- A class decorator is defined just before the class declaration, and it tells about the class behaviors.
- A class decorator is applied to the constructor of the class.
- A class decorator can be used to observe, modify, or replace a class definition.

Example of Class Decorators:

@sealed

```
class Person {  
    msg: string;  
    constructor(message: string) {  
        this.msg = message;  
    }  
    show()  
    {  
        return "Hello, " + this.msg;  
    }  
}
```

Method Decorators

- A Method Decorator is defined just before a method declaration.
- It is applied to a property descriptor for the method. It can be used to observe, modify, or replace a method definition.
- We cannot use method decorator in a declaration file.
- The expression for the method decorator function accepts three arguments.

Example of Method Decorators

```
class Item {  
    itemArr: Array;  
    constructor() {  
        this.itemArr = [];  
    }  
    @log  
    Add(item: string): void {  
        this.itemArr.push(item);  
    }  
    GetAll(): Array {  
        return this.itemArr;  
    }  
}
```

Accessor Decorators

- An Accessor Decorator is defined just before an accessor declaration.
- It is applied to the property descriptor for the accessor.
- It can be used to observe, modify, or replace an accessor's definitions.
- An accessor is a getter and setter property of the class declaration.

Example of Accessor Decorators

```
class Employee {  
    private _salary: number;  
    private _name: string;  
  
    @configurable(false)  
    get salary() { return 'Rs. ${this._salary}'; }  
    set salary(salary: any) { this._salary = +salary; }  
  
    @configurable(true)  
    get name() { return 'Sir/Madam, ${this._name}'; }  
    set name(name: string) { this._name = name; }  
}
```

Property Decorators

- A property decorator is defined just before a property declaration.
- It is similar to the method decorators.
- The only difference between property decorators and method decorators is that they do not accept property descriptor as an argument and do not return anything.

Example of Property Decorators

```
class Company {  
    @ReadOnly  
    name: string = "NIET";  
}  
  
let comp = new Company();  
comp.name = 'SSSIT.com'; console.log(comp.name);
```

Parameter Decorators

- A parameter decorator is defined just before a parameter declaration.
- It is applied to the function for a class constructor or method declaration.
- It cannot be used in a declaration file or in any other ambient context (such as in a declared class).

Example of Parameter Decorators

```
class Person {  
    msg: string;  
    constructor(message: string) {  
        this.msg = message;  
    }  
    @validate  
    show(@required name: string) {  
        return "Hello " + name + ", " + this.msg;  
    }  
}
```

Q 1 Using what can we modify the behaviour of a class?

- a. party decorators
- b. property decorators
- c. class modifier
- d. class decorators

Q 2 Which of these is a backported feature of TypeScript?

- e. arrow
- f. methods
- g. modules
- h. classes

Q 3 A Decorator is a special kind of declaration that can be applied to classes, methods, accessor, property, or parameter(True/False)

- A. True
- B. False

Q 4 Decorators are simply functions that are prefixed expression symbol

- C. #
- D. &
- E. @

Q 5 Decorators are an experimental feature proposed for ES6(True/False)

- F. True
- G. False

Q 6 A class decorator is defined just before the class declaration, and it tells about the method behaviors(True/False)

- A. True
- B. False

Q 7 A class decorator is applied to the ----- of the class.

- C. Method
- D. Constructor
- E. Variable

Q 8 Method Decorators is applied to a ----- descriptor for the method

- F. Parameter
- G. Method
- H. Property

Q 9 The expression for the method decorator function accepts----- arguments.

- I. One
- J. Two
- K. Three
- L. Four

Q 10 An Accessor Decorator is defined just before a mutator declaration (True/False)

- M. True
- N. False

Daily Quiz(Cont..)

Q 11 An accessor is a getter and setter property of the class declaration.(True/False)

- A. True
- B. False

Q 12 Property Decorator is similar to the----- decorators

- C. Class
- D. Method
- E. Accessor

Q 13 The only difference between property decorators and method decorators is that they do not accept property descriptor as an argument and always return anything(True/False)

- F. True
- G. False

Q 14 Parameter Decorator is applied to the function for a class constructor or method declaration. (True/False)

- H. True
- I. False

Q 15 A parameter decorator is defined just before a----- declaration.

- J. Class
- K. Method
- L. Property
- M. Parameter

- ES6 introduced a new operator referred to as a spread operator, which consists of three dots (...).
- It allows an iterable to expand in places where more than zero arguments are expected.
- It gives us the privilege to obtain the parameters from an array.

Syntax:

`var variablename1 = [...value];`

- The three dots (...) in the above syntax are the spread operator, which targets the entire values in the particular variable.

Example

```
let colors = ['Red', 'Yellow'];  
let newColors = [...colors, 'Violet', 'Orange', 'Green'];  
console.log(newColors);
```

Output

```
[ 'Red', 'Yellow', 'Violet', 'Orange', 'Green' ]
```

Without using spread operator

```
let colors = ['Red', 'Yellow'];  
let newColors = colors;  
newColors.push('Green');  
console.log(newColors);  
console.log(colors);
```

Output:

```
[ 'Red', 'Yellow', 'Green' ]  
[ 'Red', 'Yellow', 'Green' ]
```

Using spread operator

```
let colors = ['Red', 'Yellow'];  
let newColors = [...colors];  
newColors.push('Green');  
console.log(newColors);  
console.log(colors);
```

Output

```
[ 'Red', 'Yellow', 'Green' ]  
[ 'Red', 'Yellow' ]
```

Q 1 The term 'Spread Operator' can be defined as an operator that allows expending the array into separate elements.

- A. Two Dot(..)
- B. Three Dot(...)
- C. Four Dot(....)

Q 2 The Spread Operator is the concepts of-----

- D. Java Script
- E. Type Script
- F. ES6
- G. ES7

Q 3 Spread Operator allows an iterable to expand in places where more than zero arguments are expected(True/False)

- H. True
- I. B. False

Q 4 Spread Operator does not give us the privilege to obtain the parameters from an array.(True/False)

- J. True
- K. False

Q 5 Spread operator does not make changes in the original array it also does operations in the spread operator(True/False)

- L. True
- M. False

Daily Quiz(Cont..)

Q 6 Spread Operator is a very simple and powerful feature introduced in the ES6 standard of-----

- A. VBScript
- B. TypeScript
- C. Node.JS
- D. Java Script

Q 7 The Spread operator allows an iterable to expand in places where 0+ arguments are expected. (True/False)

- E. True
- F. False

Q 8 Spread operator is mostly used in the variable array where there is more than 1 value is expected. (Right/Wrong)

- G. Right
- H. Wrong

Q 9 Spread operator can not be used in many cases, like when we want to expand, copy, concat, with math object(True/False)

- I. True
- J. False

Q 10 Spread operator syntax is similar to the-----parameter

- K. Expand
- L. Iterate
- M. rest

- The Typescript has two operators for checking equality.
- One is == (equality operator or loose equality operator) and the other one is === (strict equality operator).
- Both of these operators check the value of operands for equality.
- But, the difference between == & === is that the == does a type conversion before checking for equality.

Two values are equal if they are

- identical strings
- numerically equivalent numbers
- identical Boolean values
- the same object (reference types)

Example:

```
let a=10
```

```
let b=10
```

```
console.log(a==b) //true
```

```
console.log(a===b) //true
```


Difference between == & ===

- If types are same then there is no difference between == & ===
- If types are different then == does a type conversion.
- It will attempt to convert them to a string, number, or boolean. before doing the comparison.
- === returns false.

Equality Operator ==

```
let a=10
```

```
let b=10
```

```
console.log(a==b) //true
```

But, in the following code, the variable b is a string and not a number.

The Typescript makes the type conversion of b from string to number and then does the comparison. Hence the result is true again.

```
let a=10
```

```
let b="10"
```

```
console.log(a==b) //true
```

```
let a="01"
```

```
let b=1
```

```
console.log(a==b);//true
```

Strict Equality Operator ===

- The strict Equality operator, returns false if the types are different.

```
let a=10
```

```
let b=10
```

```
console.log(a===b) //true
```

```
let a=10
```

```
let b="10"
```

```
console.log(a===b) //false
```

Q 1 What return the following code?

```
let a=10
```

```
let b=10
```

```
console.log(a==b)
```

```
console.log(a===b)
```

A. true

B. false

C. Compile Error

Q 2 If types are same then there is no difference between == & ===(True/False)

D. True

E. False

Q 3 What return the following code?

```
let a="01"
```

```
let b=1
```

```
console.log(a==b)
```

F. true

G. false

H. Compile Error

Q 4 What return the following code?

```
let a="10"
```

```
let b=10
```

```
console.log(a===b)
```

I. true

J. false

K. Compile Error

Q 5 What return the following code?

```
let a=20
```

```
let b=20
```

```
console.log(a===b)
```

- A. true
- B. false
- C. Compile Error

Q 6 Equality Operator does not check the type of the operand. It tries to convert them to string, number, or Boolean. (True/False)

- D. True
- E. False

Q 7 The strict Equality operator, returns false if the types are different.(True/False)

- F. True
- G. False

Q 8 The output of console.log(NaN==NaN); is:

- H. true
- I. false

Q 9 The output of console.log(-0==0); is:

- J. true
- K. false

Q 10 The output of console.log(null==undefined); is:

- L. true
- M. false

- ❑ JavaScript **ES6** (also known as **ECMAScript 2015** or **ECMAScript 6**) is the newer version of JavaScript that was introduced in 2015
- ❑ ECMAScript 2015 or ES2015 is a significant update to the JavaScript programming language. It is the first major update to the language since ES5 which was standardized in 2009. Therefore, ES2015 is often called ES6.
- ❑ ECMAScript is the standard that JavaScript programming language uses. ECMAScript provides the specification on how JavaScript programming language should work.

JavaScript let

JavaScript let is used to declare variables. Previously, variables were declared using the var keyword

The variables declared using let are block-scoped. This means they are only accessible within a particular block. For example,

```
// variable declared using let|
let name = 'Sara';
{
  // can be accessed only inside
  let name = 'Peter';

  console.log(name); // Peter
}
console.log(name); // Sara
```

JavaScript const

The const statement is used to declare constants in JavaScript. For example,

```
// name declared with const cannot be changed  
const name = 'Sara';
```

Once declared, you cannot change the value of a const variable

JavaScript Arrow Function

In the ES6 version, you can use arrow functions to create function expressions. For example,

This function

```
// function expression  
let x = function(x, y) {  
    return x * y;  
}
```

can be written as

```
// function expression using arrow function  
let x = (x, y) => x * y;
```

JavaScript Classes

JavaScript class is used to create an object. Class is like a constructor function. For example,

```
class Person {  
  constructor(name) {  
    this.name = name;  
  }  
}
```

Keyword class is used to create a class. The properties are assigned in a constructor function. Now you can create an object.

```
class Person {  
  constructor(name) {  
    this.name = name;  
  }  
}  
  
const person1 = new Person('John');  
  
console.log(person1.name); // John
```

Asynchronous Programming in ES6

- ES6 makes asynchronous programming easier with the `async` and `await` keywords.
- The `async...await` syntax in ES6 offers a new way write more readable and scalable code to handle promises.
- It uses the same features that were already built into JavaScript.
- `Promises.all()` provides a means of corralling several asynchronous methods into an array of results that we can then iterate over for additional processing.

Q 1 ECMAScript is a _____ Standard

- A. Language
- B. Java Script
- C. Programming

Q 2 ES6 is the implementation of-----

- D. Java Script
- E. Type Script
- F. VBScript
- G. ECMAScript

Q 3 ES6 is officially called _____

- H. Java Script 2015
- I. VBScript 2016
- J. ECMAScript 2015

Q 4 ES6 can be used for _____

- K. Client-side
- L. Server-Side
- M. Both client side and server side scription

Q 5 During destructuring, you can either declare variables or assign to them, or both(True/False)

- N. True
- O. False

Q 6 ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript(True/False)

- P. True
- Q. False

Daily Quiz(Cont..)

Q 7 ES6 makes asynchronous programming easier with the async keyword only(True/False)

- A. True
- B. False

Q 8 ES6 uses the same features that were already built into JavaScript(True/False)

- C. True
- D. False

Q 9 Promises.all() provides a means of corralling several asynchronous methods into an array of results that we can then iterate over for additional processing.(True/False)

- E. True
- F. False

Q 10 JavaScript ES6 is nothing but the 6th version of JavaScript introduced in 2015(Right/Wrong)

- G. Right
- H. Wrong

Q 11 It is mandatory to use the function keyword for defining functions in ES6(True/False)

- I. True
- J. False

Q 12 It is also mandatory to use the return keyword for defining functions in ES6(True/False)

- K. True
- L. False

Q 13 ES6 shows relatively higher performance.(True/False)

- M. True
- N. False

- A Promise represents something that is eventually fulfilled.
- A Promise can either be rejected or resolved based on the operation outcome.
- ES6 Promise is the easiest way to work with asynchronous programming in JavaScript.
- Asynchronous programming includes the running of processes individually from the main thread and notifies the main thread when it gets complete.
- Prior to the Promises, Callbacks were used to perform asynchronous programming.

- A Callback is a way to handle the function execution after the completion of the execution of another function.
- A Callback would be helpful in working with events. In Callback, a function can be passed as a parameter to another function.

Creating a Promise

- In JavaScript, we can create a Promise by using the Promise() constructor.

Syntax

- `const Promise = new Promise((resolve,reject)
=> {...});`

Example of Promise Constructor

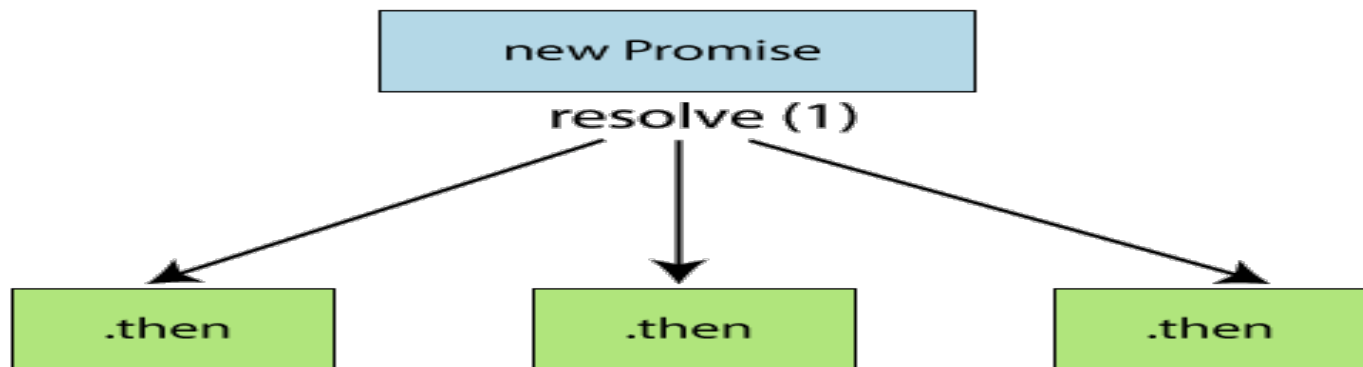
```
let Promise = new Promise((resolve, reject)=>{
  let a = 3;
  if(a==3){
    resolve('Success');
  }
  else{
    reject('Failed');
  }
})
Promise.then((message)=>{
  console.log("It is then block. The message is: ?+ message)
}).catch((message)=>{
  console.log("It is Catch block. The message is: ?+ message)
})
```

Output

It is then block. The message is: Success

Promise with Chaining

- Promise chaining allows us to control the flow of JavaScript asynchronous operations.
- By using Promise chaining, we can use the returned value of a Promise as the input to another asynchronous operation



Promise with Chaining(cont..)

- Sometimes, it is desirable to chain Promises together.
- For example, suppose we have several asynchronous operations to be performed. When one operation gives us data, we will start doing another operation on that piece of data and so on.
- Promise chaining is helpful when we have multiple interdependent asynchronous functions, and each of these functions should run one after another.

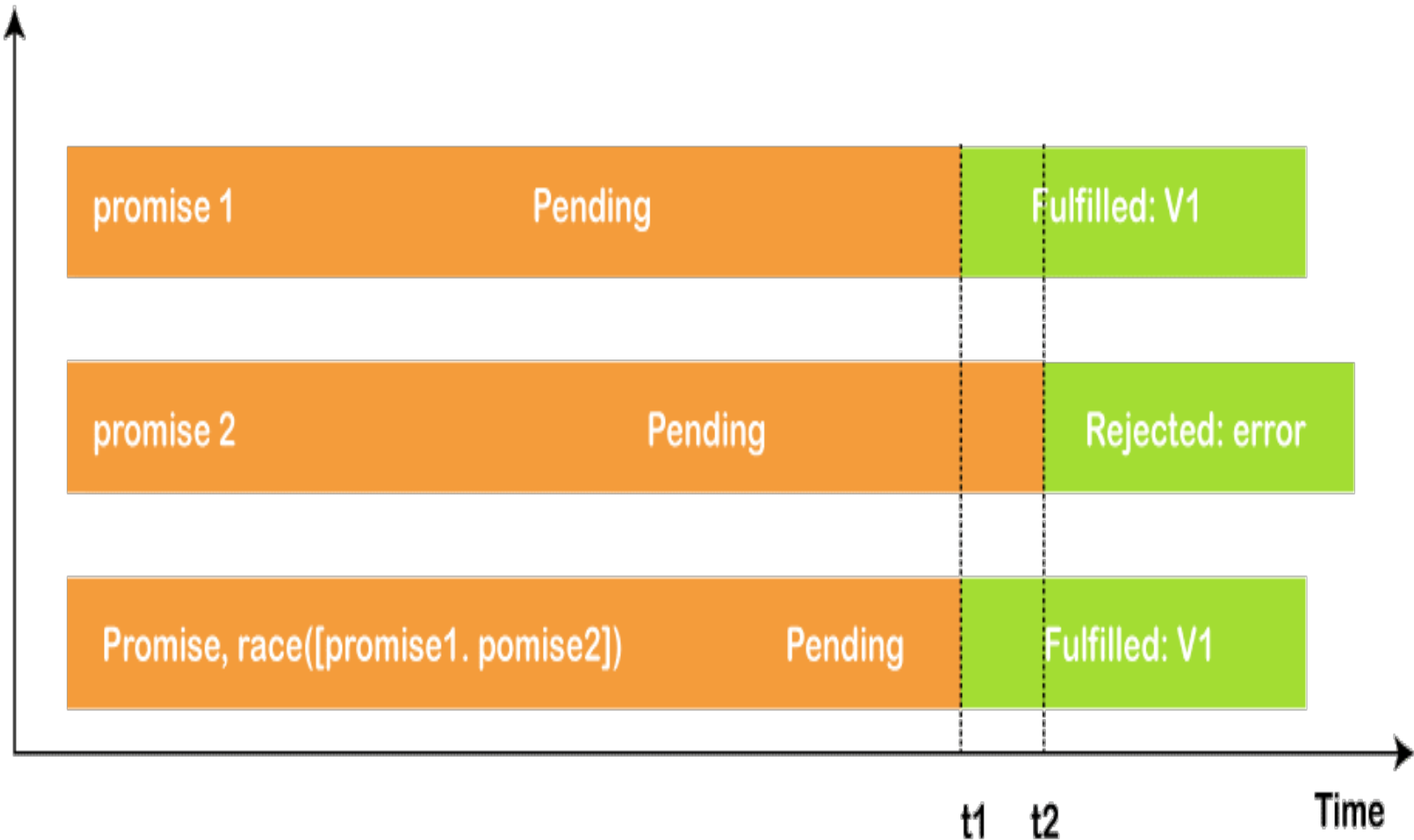
Example:

```
<script>
let promise_variable = new Promise((resolve, reject) => {
  resolve("JavaScript chaining");
});
promise_variable
  .then( function (resultInfo1){
    console.log(resultInfo1);
    return new Promise((resolve,reject) =>{
      resolve("JavaTpoint");
    })
  })
  .then((resultInfo2) => {
    console.log(resultInfo2);
  });
</script>
```

Promise Race Method

- The `Promise.race()` method delivers a promise that is fulfilled or rejected along with the value or justification from each promise in an iterable as soon as that promise is fulfilled or rejected.
- Any promise successfully fulfilled or rejected at the outset will be carried out first.
- The results of the other promises won't be shown as an output, so we might think of this particular method in the form of a real-life example where several people are running in a race, and whoever wins comes first wins the race.

The promise.race() working Diagram



The promise.race() Example

```
<script>
const pr1 = new Promise((resolve, reject) => {
  setTimeout(resolve, 700, "first promise");
});
const pr2 = new Promise((resolve, reject) => {
  setTimeout(resolve, 500, "second promise");
});
Promise.race([pr1, pr2]).then((value) => {
  document.getElementById('data').innerHTML = value;
});
</script>
```

Daily Quiz(Cont..)

Q 1 Promises are a way to implement asynchronous programming in JavaScript(ES6 which is also known as ECMAScript-6). (True/False)

- A. True
- B. False

Q 2 A Promise acts as a ----- for future values

- C. Component
- D. Container
- E. Property

Q 3 A Promise can either be rejected or resolved based on the operation outcome.(True/False)

- F. True
- G. False

Q 4 A -----programming includes the running of processes individually from the main thread and notifies the main thread when it gets complete

- H. Synchronous
- I. Asynchronous
- J. Procedural

Q 5 In ES6, we can create a Promise by using the Promise() constructor.(True/False)

- K. True
- L. False

Daily Quiz(Cont..)

Q 6 A ----- is a way to handle the function execution after the completion of the execution of another function.

- A. Setback
- B. Rollback
- C. Callback

Q 7 Promise chaining allows us to control the flow of JavaScript ----- operations.

- D. Callback
- E. Setback
- F. Asynchronous
- G. Synchronous

Q 8 By using Promise chaining, we can use the returned value of a Promise as the input to another asynchronous operation(True/False)

- H. True
- I. False

Q 9 A Promise is created when we are sure of whether or not the assigned task will be completed(True/False)

- J. True
- K. False

Q 10 A Promise is not always in one of the following states:

- a. fulfilled:
- b. rejected:
- c. pending:
- d. Constructed:

Q 11 Promise chaining is helpful when we have multiple interdependent synchronous functions(True/False)

- A. True
- B. False

Q 12 The----- method delivers a promise that is fulfilled or rejected along with the value or justification from each promise in an iterable as soon as that promise is fulfilled or rejected

- C. Promise.all()
- D. Promise.do()
- E. Promise.race()

Q 13 The ----- static method accepts a list or iterable of Promises(e.g.: Array) and returns a new Promise that resolves when all Promise in the iterable has resolved or rejects if at least one of the promises in the iterable has rejected.

- F. Promise.race()
- G. Promise.all()
- H. Promise.reject()
- I. Promise.accept()

Daily Quiz(Cont..)

Q 14 -----is a simple concept by which we may initialize another promise inside our .then() method and accordingly we may execute our results.

- A. Constructor Chaining
- B. Promise Chaining
- C. Destructive Chaining

Q 15 A Promise is executed by using the .then() method written after the declared promise.(True/False)

- D. True
- E. False

Q 16 Any promise successfully fulfilled or rejected at the ----- will be carried out first.

- F. Inset
- G. Outset
- H. Callback

Q 17 The results of the other promises won't be shown as an output, so we might think of this particular method in the form of a real-life example where several people are running in a race, and whoever wins comes first wins the race.(True/False)

- I. True
- J. False
- K. Can Not say

Daily Quiz(Cont..)

Q 18 The second one is reject() which is responsible for the unsuccessful completion of an operation, and we can pass on text inside it.(True/False)

- A. True
- B. False

Q 19 Promise chaining is helpful when we have multiple interdependent asynchronous functions(True/False)

- a. True
- b. False

Q 20 We can declare the promise by using the following syntax.

```
let promise = new Promise((resolve , reject)=>{  
    resolve('Hello JavaScript !');  
});
```

- A. True
- B. False

Weekly Assignment

1. Where can we place JavaScript ? **[CO4]**
2. Explain screen output and keyboard input method in JavaScript. **[CO4]**
3. Define instance of operator in JavaScript. **[CO4]**
4. Explain various datatypes used in JavaScript. **[CO4]**
5. Write a JavaScript to find factorial of a number. **[CO4]**
6. What is a DTD ? Mention its types. **[CO4]**
7. What are composite datatypes in Java Script ? **[CO4]**
8. What are the different levels of headings in HTML ? **[CO4]**
9. How to create date object in JavaScript ? Explain any four methods of JavaScript date object. **[CO4]**
10. What are the different ways that we can include JavaScript to a XHTML document. **[CO4]**

Q 1 Which of the following methods/operation does JavaScript use instead of == and !=?

- a) JavaScript uses equalto()
- b) b) JavaScript uses equals() and notequals() instead
- c) c) JavaScript uses bitwise checking
- d) d) JavaScript uses === and !== instead

Q 2 Why is JavaScript Engine needed?

- e) Both Compiling & Interpreting the JavaScript
- f) b) Parsing the JavaScript
- g) c) Interpreting the JavaScript
- h) d) Compiling the JavaScript

Q 3 What would be the result of 2+5+"3"?

- i) 5
- j) 7
- k) 73
- l) 37

Q 4. Java script is an _____ language?

- (A) object oriented
- (b) object base
- © procedural
- (d) None of these

Q 5. Which of the following keywords is used to define a variable in JavaScript?

- (a) let
- (b) var
- (C) both A & B
- (d) None of the above

Q 6. How can a datatype be declared to be constant type ?

- (A) var
- (b) const
- © let
- (D) constant

Q 7 .Java script is ideal to

- (A)make computations in HTML simpler
- (B) minimize storage requirements on the web server
- (c) increase the download time for the client
- (D) none of the mentioned

Q 8 .The basic difference between javascript & java is

- (A)There is no difference
- (B). Functions are considered as fields
- ©. Variables are specific
- (D). Functions are values, and there is no hard distinction between methods and fields

Q 9 . Which of the following is not javascript framework?

- (A) Node
- (B) Vue
- (C) React
- (D) Cassandra

Q 10. Arrays in JavaScript are defined by which of the following statements?

- a) It is an ordered list of values
- b) It is an ordered list of objects
- c) It is an ordered list of string
- d) It is an ordered list of functions

Q 11. Which of the following is not JavaScript data types?

- e) Null type
- f) Undefined type
- g) Number type
- h) All of the mentioned

Q 12. Where is Client-side JavaScript code is embedded within HTML documents?

- a) A URL that uses the special javascript:code
- b) A URL that uses the special javascript:protocol
- c) A URL that uses the special javascript:encoding
- d) A URL that uses the special javascript:stack

Q 13. Name some of the JavaScript Frameworks?

- (A) Angular
- (B) React
- (C) Vue
- (D) All of them

Q 14 In how many ways a JavaScript code can be involved in an HTML file?

- (E) Inline
- (F) Internal
- (G) External
- (H) All of them

Q 15 List out the different ways an HTML element can be accessed in a JavaScript code.

- (i) `getElementById('idname')`: Gets an element by its ID name
- (ii) `getElementsByClass('classname')`: Gets all the elements that have the given class name.
- (iii) `getElementsByTagName('tagname')`: Gets all the elements that have the given tag name.
- (iv) `querySelector()`: This function takes CSS style selector and returns the first selected element.
- (v) All of them.

Q 16 For largest level of heading which tag is used?

- A) <h1>
- B) <h6>
- C) <heading>**
- D) <head>

Q 17 API stands for

- a)Application Programming Interface
- b) Application Programming Index
- c) Applet Programming Interface
- (d) None of these

Q 18 For handling user interaction ____side scripting is useful

- Server
- b)Client
- c)Server & Client
- d) None of these

Q 19 What is the prototype represents in the following JavaScript code snippet?

```
function javascript() {};
```

- a) Not valid
- b) Prototype of a function
- c) Function javascript
- d) A custom constructor

Q 20 Why event handlers is needed in JS?

- a) Allows JavaScript code to alter the behaviour of windows
- b) Adds innerHTML page to the code
- c) Change the server location
- d) Performs handling of exceptions and occurrences

Glossary Questions

JAD stands for

- a) Joint Application Development
- b) Joint Application Difference
- c) Joint Application Demo
- d) none of these

Java script is scripting language introduce by

- e) IE
- f) Microsoft
- g) Netscape
- h) SunJava

___ operator calculates the remainder by dividing two integers.

- i) / b) rem c) ^ d) %

Glossary Questions

Which tag is used for smallest heading protocol stack.

Marquee tag has following attributes

- a) behavior
- b) direction
- c) position
- d) both a and b.

The word internet stands for _____

- a) International Network
- b) Internal Network
- c) Intermediate Network
- d) none of these

_____ is used to create a new paragraph

- (a) <from> (b) (c) <p> (d) <marquee>

Sessional Papers

Printed page: 2

Subject Code: ACSE0505

Roll No:

--	--	--	--	--	--	--	--	--	--	--	--	--

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)

Affiliated to Dr. A.P. J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow

Course: B.Tech.

Branch: AI/IoT/CS/AIML/DS/IT/CSE/M.Tech(Int)

Semester: V

Sessional Examination: First Sessional

Subject Name: Web Technology

Year- (2022 - 2023)

Time: 1.15Hours

Max. Marks:30

General Instructions:

- This Question paper consists of 02 pages & 05 questions. It comprises three Sections -A, B, & C. You are expected to answer them as directed.
- Section A - Q.No- 1 is of one 1 mark each & Q.No- 2 carries 2 mark each.
- Section B - Q. No- 3 carries 5 marks each.
- Section C - Q.No-4 & 5 carries 6 mark each. Attempt any one part a or b

SECTION – A

[08Marks]

- | | |
|--|--------------------|
| 1. All questions are compulsory- | (4×1=4) |
| a. Write the full form of DOM | (1) CO1 |
| a. Document object model | |
| b. Digital object model | |
| c. Design orientation model | |
| d. Development object model | |
|
b. Father of Web..... and birth of web..... |
(1) CO1 |
| a. Pascal, 1989 | |
| b. Tim Berners Lee, 1960 | |
| c. Bob, 1989 | |
| d. Tim Berners Lee, 1989 | |
|
c. Which of following is frontend development |
(1) CO2 |
| a. Java | |
| b. PHP | |
| c. XML | |
| d. All of these. | |
|
d. In web development HTML used for design in |
(1) CO2 |
| a. Server | |

- b. Browser
- c. Web Page
- d. None of the above

2. All questions are compulsory- (2×2=4)
- a. What do you understand by client-server computing. (2) CO2
 - b. Describe the term W3C . (2) CO1

SECTION – B

[10Marks]

3. Answer any two of the following- (2×5=10)
- a. Discuss the basic principle involved in developing a web site, types of web site. (5) CO1
 - b. Write the short notes on history of Web and Internet. (5) CO1
 - c. Describe the concept of HTML structure. (5) CO2

SECTION – C

[12Marks]

- 4 Answer any one of the following- (1×6=6)
- a. Write the different protocols used in governing web. (6) CO2
 - b. Write the short notes on web hosting and types of web hosting packages. (6) CO1

Subject Code: ACSBS0513

Sessional Papers

	b.	Write a program to implement interface that calculate the area of rectangle using abstract method declaration within interface	(5)	1	K5
	c.	Explain the process to create a package in JAVA? Give suitable example	(5)	1	K4
<u>SECTION – C</u>					
4.	Answer any <u>one</u> of the following-		[2×6=12]	CO	
	a.	Explain the various types of Exception hierarchy in Java. Write a program to implement <u>ArrayIndexOutOfBoundsException</u> in Java.	(6)	1	K5
	b.	List various ways to create thread in Java? Write a program to generate even and odd thread by using these ways.	(6)	2	K5
5.	Answer any <u>one</u> of the following-		[2×6=12]	CO	
	a.	Design an interface inside the class and interface with suitable example	(6)	2	K6
	b.	Differentiate abstract class and interface with example.	(6)	1	K4

Old Question Paper

B.TECH.
(SEM VI) THEORY EXAMINATION 2018-19
WEB TECHNOLOGIES

Time: 3 Hours

Total Marks: 70

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief. 2 x 7 = 14

- (a) Explain the difference between Portal and Website. (Unit-1)
- (b) Packages and Interfaces both acts as a container, How? (Unit-1)
- (c) How XML is different from HTML? (Unit-2)
- (d) What is the difference between <div> and tag in HTML? (Unit-2)
- (e) We want to submit the data on page itself, what can be done? (Unit- 4)
- (f) What is the role of CGIs in web design? (Unit-3)
- (g) Differentiate the terms SendRquest() from Hyperlink. (Unit-5)

SECTION B

2. Attempt any *three* of the following: 7 x 3 = 21

- (a) Why early planning is useful to develop an effective website? Give proper example in favor of your reason. (Unit-1)
- (b) Describe the role and importance of CSS in web designing. Also Differentiate Class and Id in CSS. (Unit-2)

Old Question Paper(cont..)

- example in favor of your reason. (Unit-1)
- (b) Describe the role and importance of CSS in web designing. Also Differentiate Class and Id in CSS. (Unit-2)
 - (c) What do you mean by Session Bean? Explain its types using suitable example. (Unit- 4)
 - (d) Explain Request Dispatcher. Also Describe different ways to get the object of request dispatcher. (Unit-5)
 - (e) Explain the role of Java Script to develop a web page. Write a java script function to check a textbox is either empty or not. (Unit-3)

SECTION C

3. Attempt any *one* part of the following: 7 x 1 = 7

- (a) Describe the objective of any Website. Which type of essential skills required being a member of web project team? (Unit-1)
- (b) Explain the difference between JSP includes directive elements & JSP includes action elements. (Unit-5)

4. Attempt any *one* part of the following: 7 x 1 = 7

- (a) What is Java Bean exactly? Write down the steps to create Java Bean. What is the role of introspection in Java Bean? (Unit-4)
- (b) What do you mean by Database Drivers, explain each type? Also explain the steps to get any value into database. (Unit-4)

Old Question Paper(cont..)

Printed Pages: 02

Sub Code: RIT601

5. Attempt any *one* part of the following:

7 x 1 = 7

- (a) “Document Type Definition (DTD) in XML is necessary”, justify the statement with suitable example. Under which conditions which DTD is more preferable? (Unit-2)
- (b) Using a frameset, create an HTML document like following:-

Header.html	
Menu.html	Output.html

Also host it as a web page on any server. (Unit-2)

6. Attempt any *one* part of the following:

7 x 1 = 7

- (a) JSP is an extension of Servlets not replacement, Justify? What problems of Servlets technology can JSP is suppose to solve? (Unit-5)
- (b) How we deploy the Servlets on Tomcat Web Container? Also explain how we change the default port number of Tomcat Container? (Unit-5)

7. Attempt any *one* part of the following:

7 x 1 = 7

- (a) Can we do session handling in HTML Page with the help of JavaScript? If yes then how or If not then why? (Unit-3)
- (b) Explain the following terms in Brief:-
 - (i) Scket Programming
 - (ii) TCP/IP Server

- Explain the importance of JavaScript. Write a java script to manipulate array
- Explain relational, logical and athematic operators.
- Explain if-else statement in java script with examples.
- Write a java Script for counting Odd, even and prime numbers from 1 to 100
- Write a java script to calculate the area of the triangle
- Write a java script to find factorial of given number.
- If $a=10$ and $b=20$ write a java script to swapping of these two numbers
- Write a script to display sum of first 10 numbers
- write a java script to validate phone number

- Write a Java Script to calculate the average of three numbers.
- Write a java script for finding the greater number between two numbers.
- Write a Java Script for on change event.
- Which java script operator calculates the remainder by dividing two integers.
- Which tag is used for smallest heading .
- which tag is used to left align the content is table cell.
- which tag is used to left align the content is table cell.

- Burdman, Jessica, “Collaborative Web Development” Addison Wesley
- Xavier, C, “ Web Technology and Design” , New Age International
- Ivan Bayross,” HTML, DHTML, Java Script, Perl & CGI”, BPB Publication

- ❖ Covered What is JavaScript how JavaScript is different than HTML,CSS.
- ❖ Covered variables, constant, keywords, operators etc.
- ❖ Discussed various java script objects, function , function parameters, function arguments etc.
- ❖ Covered JS script loop, js popup boxes, JS event etc.
- ❖ Discussed JS array, strings, strings method etc.

NPTEL video / Other study links

Unit 1	https://youtu.be/96xF9phMsWA
	https://youtu.be/Zopo5C79m2k
	https://youtu.be/Zlils7jHi1s
	https://youtu.be/htbY9-yggB0
Unit 2	https://youtu.be/vHmUVQKXIVo
	https://youtu.be/qz0aGYrrlhU
Unit 3	https://youtu.be/1Rs2ND1ryYc
Unit 4	https://youtu.be/-qfEOE4vtxE
	https://youtu.be/PkZNo7MFNFg
Unit 5	https://youtu.be/_GMEqhUyyFM

Thank You