Here are the questions with their options and the correct answers:

1. \*\*Which of the following is true about let and const in terms of scope?\*\*

- a) Both are globally scoped.

- b) Both are function-scoped.

- c) Both are block-scoped. \*\*(Correct Answer)\*\*

- d) let is block-scoped and const is function-scoped.

2. \*\*Which npm command installs both dependencies and dev dependencies?\*\*

- a) npm install --only=prod

- b) npm install --only=dev

- c) npm install \*\*(Correct Answer)\*\*

- d) npm install --save-dev

3. \*\*What will the following TypeScript code output?\*\*

```typescript

let sum = 0;

let n = 5;

while (n > 0) {

sum += n;

n--;

}

console.log(sum);

```

- a) 15 \*\*(Correct Answer)\*\*

- b) 10

- c) 5

- d) 0

4. \*\*What does the following TypeScript code output?\*\*

```typescript

let obj = {x: 10, y: 20};

for (let key in obj) {

console.log(obj[key]);

}

```

- a) x y

- b) 10 20 \*\*(Correct Answer)\*\*

- c) undefined undefined

- d) x 10 y 20

5. \*\*Which of the following statements correctly imports a default export in TypeScript?\*\*

- a) import { default } from './module';

- b) import default from './module';

- c) import (module) from './module';

- d) import myModule from './module'; \*\*(Correct Answer)\*\*

6. \*\*Consider the following type alias:\*\*

```typescript

type Point = { x: number; y: number; };

```

\*\*Which of the following is correct?\*\*

- a) let p: Point = { x: ‘10’, y: 20 };

- b) let p: Point = { x = 10, y = 20};

- c) let p: Point = { x: 10, y: 20 }; \*\*(Correct Answer)\*\*

- d) let p: Point = { x = ‘10’, y = 20 };

7. \*\*In TypeScript, which type system is used by default?\*\*

- a) Nominal typing

- b) Structural typing \*\*(Correct Answer)\*\*

- c) Hybrid typing

- d) Literal typing

8. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function myFunc(data1: number, data2: string): number {

const num: number = data2 as unknown as number;

const result: number = data1 + num;

return result;

}

const myAnswer: number = myFunc(10, "5");

console.log(myAnswer);

```

- a) Type Error

- b) 15 \*\*(Correct Answer)\*\*

- c) undefined

- d) 105

9. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

const myName = (fName: string, lName?: string): string => {

const myFullName: string = lName ? fName + " " + lName : fName;

return myFullName;

}

const myAnswer1: string = myName("Muhammad");

const myAnswer2: string = myName("Muhammad", "Fasih");

console.log(myAnswer2);

console.log(myAnswer1);

```

- a) Muhammad, Muhammad Fasih

- b) undefined, Muhammad Fasih

- c) Muhammad, undefined

- d) Muhammad Fasih, Muhammad \*\*(Correct Answer)\*\*

10. \*\*Which of the following is true about the unknown type in TypeScript?\*\*

- a) unknown can be assigned to any type without type-checking.

- b) unknown is more specific than any and requires type-checking before use. \*\*(Correct Answer)\*\*

- c) unknown and any are interchangeable.

- d) unknown represents the type of values that never occur.

11. \*\*Enums and Const Enums: What is the main advantage of using const enum over enum in TypeScript?\*\*

- a) const enum allows for runtime changes to the enum values.

- b) const enum can be used only with numeric values

- c) const enum can store string values.

- d) const enum is more memory-efficient because it is inlined at compile time. \*\*(Correct Answer)\*\*

12. \*\*Which of the following is NOT a characteristic of asynchronous programming in JavaScript?\*\*

- a) It allows for non-blocking code execution.

- b) It executes code sequentially in the order it appears. \*\*(Correct Answer)\*\*

- c) It uses callbacks, promises, and async/await to handle asynchronous operations.

- d) It can improve performance by preventing the blocking of the main thread.

13. \*\*What is the primary function of the call stack in JavaScript?\*\*

- a) To manage asynchronous operations.

- b) To keep track of function calls and their execution context. \*\*(Correct Answer)\*\*

- c) To handle memory allocation for variables.

- d) To execute code in parallel.

14. \*\*What will be the output of the following code?\*\*

```javascript

console.log("One");

setTimeout(function(){

console.log("I'am a programmer");

}, 2000)

console.log("Two");

console.log("Three");

```

- a) One , Two , Three \*\*(Correct Answer)\*\*

- b) One , Two , I am a programmer , Three

- c) I am a programmer

- d) One , Two , Three , I' am a programmer

15. \*\*What will be the output of the following code?\*\*

```javascript

const promise = new Promise((resolve, reject) => {

reject("Failure!!!");

});

promise.catch((error) => {

console.log(error);

});

```

- a) Error

- b) Failure!!! \*\*(Correct Answer)\*\*

- c) error

- d) Failure!!!! Error

16. \*\*Which of the following principles is NOT part of Object-Oriented Programming (OOP)?\*\*

- a) Encapsulation

- b) Inheritance

- c) Polymorphism

- d) Compilation \*\*(Correct Answer)\*\*

17. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

class Animal {

name: string;

constructor(name: string) {

this.name = name;

}

makeSound() {

console.log(`${this.name} makes a sound`);

}

}

class Dog extends Animal {

makeSound() {

console.log(`${this.name} barks`);

}

}

const myDog = new Dog('Rex');

myDog.makeSound();

```

- a) Rex makes a sound

- b) Rex barks \*\*(Correct Answer)\*\*

- c) Error: makeSound method not defined

- d) Error: constructor not defined

18. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function categorizeAge(age: number): string {

if (age < 13) {

return "Child";

} else if (age >= 13 && age <= 19) {

return "Teenager";

} else {

return "Adult";

}

}

let x = categorizeAge(16);

console.log(x);

```

- a) Child

- b) Teenager \*\*(Correct Answer)\*\*

- c) Adult

- d) Undefined

19. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function calculateDiscount(quantity: number): number {

let discount = 0;

if (quantity > 10) {

discount = 20;

} else if (quantity >= 5) {

discount = 10;

} else {

discount = 0;

}

return discount;

}

let x = calculateDiscount(10);

console.log(x);

```

- a) 0

- b) 10 \*\*(Correct Answer)\*\*

- c) 20

- d) 5

20. \*\*Which npm command would you use to install a package as a dev dependency?\*\*

- a) npm install <package> --save

- b) npm install <package> --global

- c) npm install <package> --save-dev \*\*(Correct Answer)\*\*

- d) npm install <package> --only=dev

21. \*\*What happens if you try to declare a const variable without initializing it?\*\*

- a) It will be automatically assigned undefined.

- b) It will throw a syntax error. \*\*(Correct Answer)\*\*

- c) It will be treated as a let variable.

- d) It will be ignored by the JavaScript engine.

22. \*\*How do you typecast a variable x to type string in TypeScript?\*\*

- a) let y: string = <string>x;

- b) let y: string = x as string;

- c) Both a and b

\*\*(Correct Answer)\*\*

- d) let y: string = x.toString();

23. \*\*Which of the following is the correct syntax for a function with a default parameter in TypeScript?\*\*

- a) function foo(x: number, y: number = 10): void { }

- b) function foo(x: number, y: 10 = number): void { }

- c) function foo(x: number, y = 10): number { }

- d) function foo(x: number, y: number): void { y = 10; } \*\*(Correct Answer)\*\*

24. \*\*What will be the result of the following TypeScript code?\*\*

```typescript

class Car {

private brand: string;

constructor(brand: string) {

this.brand = brand;

}

getBrand() {

return this.brand;

}

}

const myCar = new Car('Toyota');

console.log(myCar.getBrand());

```

- a) Car

- b) Toyota \*\*(Correct Answer)\*\*

- c) undefined

- d) Error: brand is private

25. \*\*In what order does JavaScript execute code by default?\*\*

- a) Bottom to top

- b) Left to right

- c) Top to bottom \*\*(Correct Answer)\*\*

- d) Right to left

26. \*\*What will be the output of the following code?\*\*

```javascript

console.log('A');

setTimeout(() => {

console.log('B');

}, 0);

console.log('C');

```

- a) A C B \*\*(Correct Answer)\*\*

- b) A B C

- c) B A C

- d) C B A

27. \*\*What will be the output of the following code?\*\*

```typescript

const arr: number[] = [1, 2, 3, 4, 5, 6, 7];

const result = arr.fill(5, 1, 4);

console.log(result);

```

- a) [1, 5, 5, 5, 5, 6, 7] \*\*(Correct Answer)\*\*

- b) [5, 5, 5, 5, 5, 6, 7]

- c) [1, 5, 5, 5, 5, 5, 5]

- d) [1, 5, 5, 5, 6, 7]

28. \*\*What will be the output of the following code?\*\*

```typescript

let array: number[] = [1, 2, 3, 4, 5, 6, 7, 8];

const numGreater: number[] = array.filter((CurrentEle: number) => CurrentEle > 4);

console.log(numGreater);

```

- a) [1, 2, 3, 4]

- b) [5, 6, 7, 8] \*\*(Correct Answer)\*\*

- c) [4, 5, 6, 7, 8]

- d) [5, 6, 7, 8, 9]

29. \*\*What is the output of the following code?\*\*

```typescript

async function foo() {

console.log('Start');

await setTimeout(() => {

console.log('Middle');

}, 1000);

console.log('End');

}

foo();

```

- a) Start, End, Middle \*\*(Correct Answer)\*\*

- b) Start, Middle, End

- c) End, Start, Middle

- d) Middle, Start, End

30. \*\*Given the following module `math.ts`, which statement correctly imports both the default export and a named export?\*\*

```typescript

export default class Calculator {}

export const PI = 3.14;

```

- a) import Calculator, PI from './math'

- b) import { Calculator, PI } from './math'

- c) import Calculator, { PI } from './math' \*\*(Correct Answer)\*\*

- d) import { default as Calculator, PI } from './math'

31. \*\*Given the following type alias:\*\*

```typescript

type Callback = (data: string) => void;

```

\*\*Which function signature correctly matches the alias?\*\*

- a) function cb(data: string): void {} \*\*(Correct Answer)\*\*

- b) function cb(data: number): void {}

- c) function cb(data: string): any {}

- d) function cb(data: string): string {}

32. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function checkCondition(x: boolean, y: boolean): boolean {

return !x || y;

}

let result = checkCondition(true, false);

console.log(result);

```

- a) true

- b) false \*\*(Correct Answer)\*\*

- c) undefined

- d) null

33. \*\*If a variable declared with `let` inside a block is then used outside that block, what will happen?\*\*

- a) It will be accessible outside the block.

- b) It will throw a reference error. \*\*(Correct Answer)\*\*

- c) It will be automatically hoisted to the top of the function.

- d) It will be undefined outside the block.

34. \*\*What will be the output of the following code?\*\*

```typescript

const student = {

name: "Muhammad Fasih",

age: 20,

isStudent: true,

};

const { name, age, isStudent } = student;

const myKey: keyof typeof student = age === 21 ? "name" : "age";

student["name"] = "Salman Shahid";

console.log(student[myKey]);

console.log(name);

```

- a) Salman Shahid, Muhammad Fasih

- b) Salman Shahid, Salman Shahid

- c) Muhammad Fasih, Muhammad Fasih

- d) Salman Shahid, Salman Shahid \*\*(Correct Answer)\*\*

Here are the answers to the additional questions:

35. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

type A = {

name: string;

age: number;

}

type B = {

name: string;

rollNum: number;

}

let myObj1: A = {

name: "Muhammad Fasih",

age: 20,

}

let myObj2: B = {

name: "Salman Shahid",

rollNum: 1234,

}

myObj1 = myObj2 as unknown as A;

console.log(myObj1.name === myObj2.name);

console.log(myObj1.age);

```

- a) false, 20

- b) true, 20

- c) true, undefined \*\*(Correct Answer)\*\*

- d) Error

36. \*\*What is the role of the event loop in JavaScript?\*\*

- a) To execute synchronous code.

- b) To push functions onto the Web API

- c) To manage and coordinate asynchronous tasks and their callbacks. \*\*(Correct Answer)\*\*

- d) To manage variable scopes.

37. \*\*What is the output of the following TypeScript code?\*\*

```typescript

let arr = [1, 2, 3, 4];

for (let i = 0; i < arr.length; i++) {

if (i === 2) continue;

console.log(arr[i]);

}

```

- a) 1 2 3 4

- b) 1 3 4

- c) 1 2 4 \*\*(Correct Answer)\*\*

- d) 1 2 4 3

38. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

let array = [10, 20, 30];

for (let value of array) {

console.log(value);

}

```

- a) 0 1 2

- b) 10 20 30 \*\*(Correct Answer)\*\*

- c) value

- d) [10, 20, 30]

39. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

class Person {

name: string;

age: number;

constructor(name: string, age: number) {

this.name = name;

this.age = age;

}

greet() {

console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);

}

}

const john = new Person('John', 30);

john.greet();

```

- a) Hello, my name is John and I am 30 years old. \*\*(Correct Answer)\*\*

- b) Hello, my name is undefined and I am undefined years old.

- c) Error: constructor not defined

- d) Error: greet method not defined

40. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function sum(data1: number, data2?: number) {

if(data2){

const data3: number = 10;

return data1 + data2 + data3;

}

else {

return data1 + data3;

}

}

const myAnswer1: number = sum(5);

const myAnswer2: number = sum(10,15);

console.log(myAnswer2);

console.log(myAnswer1);

```

- a) 15, 35

- b) 35, 15 \*\*(Correct Answer)\*\*

- c) 25, 5

- d) Error

41. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

let x = 10;

let y = 5;

let z = 0;

if (x > y && y > z) {

console.log("Condition met");

} else {

console.log("Condition not met");

}

```

- a) Condition met \*\*(Correct Answer)\*\*

- b) Condition not met

- c) Error

- d) Infinite loop

42. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

function checkConditions(a: boolean, b: boolean): boolean {

return !a || b && a;

}

let result = checkConditions(false, true);

console.log(result);

```

- a) true \*\*(Correct Answer)\*\*

- b) false

- c) Error

- d) undefined

43. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

class Shape {

public area(): number {

return 0;

}

}

class Circle extends Shape {

radius: number;

constructor(radius: number) {

super();

this.radius = radius;

}

public area(): number {

return Math.PI \* this.radius \* this.radius;

}

}

const circle = new Circle(5);

console.log(circle.area());

```

- a) 25

- b) 78.53981633974483 \*\*(Correct Answer)\*\*

- c) 0

- d) Error: area method not defined

44. \*\*What happens if you try to import a named export that doesn't exist in the module?\*\*

- a) The import statement will be ignored

- b) TypeScript will throw a compile-time error \*\*(Correct Answer)\*\*

- c) JavaScript will throw a runtime error

- d) The import will be silently ignored

45. \*\*Consider the following two interfaces:\*\*

```typescript

interface Dog {

bark(): void;

}

interface Cat {

bark(): void;

}

```

\*\*Which of the following statements is true in TypeScript?\*\*

- a) let pet: Dog = new Cat(); is valid due to structural typing

- b) let pet: Dog = new Cat(); is invalid due to nominal typing

- c) Dog and Cat are incompatible because they have different names

- d) Dog and Cat are compatible only if they are in the same file. \*\*(Correct Answer)\*\*

46. \*\*How can you import all exports from a module as a single object in TypeScript?\*\*

- a) import all from './module'

- b) import \* from './module'

- c) import \* as module from './module' \*\*(Correct Answer)\*\*

- d) import { all } from './module'

47. \*\*What will be the output of the following TypeScript code?\*\*

```typescript

let i = 0;

do {

i++;

} while (i < 3);

console.log(i);

```

- a) 0

- b) 1

- c) 2

- d) 3 \*\*(Correct Answer)\*\*

48. \*\*Given the following type aliases:\*\*

```typescript

type Employee = { id: number; name: string; };

type Person = { id: number; name: string; };

```

\*\*Which of the following assignments is valid in TypeScript?\*\*

- a) let emp: Employee = { id: 1, name: 'Alice' }; let p: Person = emp;

- b) let emp: Employee = { id: 1, name: 'Alice' }; let p: Person = { id: 2, name: 'Bob' }; emp = p; \*\*(Correct Answer)\*\*

- c) Both a and b

- d) Neither a nor b

49. \*\*How would you import multiple named exports from a module?\*\* //File name module.ts

```typescript

export const a = 1;

export const b = 2;

export const c = 3;

```

- a) import { a, b, c } from './module'; \*\*(Correct Answer)\*\*

- b) import a, b, c from './module';

- c) import \* as abc from './module';

- d) import { default as abc } from './module';

50. \*\*How do you create an instance of a TypeScript class?\*\*

- a) new MyClass() \*\*(Correct Answer)\*\*

- b) MyClass.new()

- c) MyClass.create()

- d) instance MyClass()