WEEK - 1 ASSESSMENT

1. What will be the output of this code?

```
console.log(x);
var x = 5;
console.log(x);
```

2. What happens when you run this code?

```
const arr = [1, 2, 3];
arr.push(4);
console.log(arr);
```

3. Fix this code to avoid hoisting issues:

```
function test() {
  console.log(name);
  var name = "Alice";
  return name;
}
```

4. What is 'this' in the arrow function?

```
const obj = {
  name: "Test",
  regular: function() { return this.name; },
  arrow: () => this.name
};
console.log(obj.arrow());
```

5. Write an arrow function that takes two numbers and returns their sum:

6. What will this output?

```
async function test() {
  console.log(1);
  await Promise.resolve();
  console.log(2);
}
console.log(3);
test();
console.log(4);
```

7. Convert this Promise chain to async/await with try/catch:

```
fetch('/api/data')
  .then(res => res.json())
  .then(data => console.log(data))
  .catch(err => console.error(err));
```

8. What will be logged?

```
function outer() {
  let count = 0;
  return function inner() {
    count++;
    return count;
  };
}
const counter = outer();
console.log(counter());
console.log(counter());
```

9. Create a function that returns a private counter (increment, decrement, getValue methods):

10. What is the value of c?

```
const arr = [1, 2, 3, 4, 5];
const [a, b, ...c] = arr;
```

11. Destructure this object to get name and age with default age of 18:
`const user = { name: "Bob" };`

12. Create a class 'ProjectA' with name property and description() method, then extend it with 'ProjectB' class that overrides description():

13. Which is the correct type annotation?
`const numbers: ??? = [1, 2, 3];`

14. Create a function that takes name (string) and age (optional number) and returns a string:

15. Define an interface 'User' with id (number), name (string), and optional email (string):

16. What's the difference between interface and type alias?

17. What does Partial<User> do?

```
interface User {
  id: number;
  name: string;
  email: string;
}
type PartialUser = Partial<User>;
```

18. Use Pick utility to create a type with only 'name' and 'email' from User interface:

```
interface User {
  id: number;
  name: string;
  email: string;
  age: number;
}
```

19. Create a function that accepts string | number and returns its length (for string) or value (for number):

20. What is the result of this intersection type?

```
type A = { name: string };
type B = { age: number };
type C = A & B;
```

21. Create a class 'Person' with private property 'age' and public method 'getAge()':