Scope and Hoisting Quiz

1. **Question:** Which of the following correctly describes the difference between `var`, `let`, and `const`?

    - a) `var` and `let` declarations are hoisted and initialized as `undefined`. `const` declarations are hoisted but not initialized.
    - b) `var` declarations are hoisted and initialized as `undefined`. `let` and `const` declarations are hoisted but not initialized.
    - c) All `var`, `let` and `const` declarations are hoisted and initialized as `undefined`.
    - d) `var` declarations are hoisted and initialized as `undefined`. `let` and `const` declarations are not hoisted.

2. **Question:** What will be the output of the following code snippet?

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

    - a) `undefined`
    - b) `5`
    - c) `null`
    - d) `ReferenceError`

3. **Question:** If a variable declared with `let` or `const` is hoisted, why can't we use it before the declaration?

    - a) Because it’s still in the “temporal dead zone”.
    - b) Because JavaScript throws an error when it sees a `let` or `const` keyword.
    - c) Because `let` and `const` are not hoisted.
    - d) Because the variable is initialized to `null`.

4. **Question:** What will be the output of the following code snippet?

   ```
   let x = 1;
   {
       let x = 2;
       console.log(x);
   }
   console.log(x);
   ```

   - a) `1 1`
   - b) `2 1`
   - c) `1 2`
   - d) `2 2`

5. **Question:** What will be the output of the following code snippet?

   ```
   const x = 1;
   function example() {
       console.log(x);
       const x = 2;
   }
   example();
   ```

   - a) `1`
   - b) `2`
   - c) `undefined`
   - d) `ReferenceError`

6. **Question:** What will be the output of the following code snippet?

```
let a = 'global';
function checkScope() {
    let a = 'local';
    function nested() {
        let a = 'nested';
        function superNested() {
            a = 'superNested';
            return a;
        }
        return superNested();
    }
    return nested();
}
console.log(checkScope());
console.log(a);
```

- a) `superNested global`
- b) `nested global`
- c) `local global`
- d) `superNested superNested`

7. **Question:** Which of the following statements about `let` and `const` are true?

    - a) Variables declared with `let` and `const` are block-scoped.
    - b) Variables declared with `let` can be updated but not re-declared in the same scope.
    - c) Variables declared with `const` must be initialized during declaration.
    - d) All of the above.