Q1. What is error handling in JavaScript? Explain the try, catch, and finally blocks with an example.

Ans: Error handling in JavaScript is the process of detecting, managing, and responding to runtime errors so that the program does not crash and can continue running smoothly.

JavaScript uses try, catch, and finally blocks for error handling.

try, catch, and finally Blocks

1. try Block

- Contains code that may cause an error.
- JavaScript attempts to execute this code.

```
try {
  // risky code
}
```

2. catch Block

- Executes only if an error occurs in the try block.
- Catches the error and allows handling it gracefully.

```
catch (error) {
  // handle error
}
```

3. finally Block

- Executes whether an error occurs or not.
- Used for cleanup tasks.

```
finally {
  // always executed
}
```

Example:

```
try {
  let x = 10;
  let y = x / z; // z is not defined
  console.log(y);
} catch (error) {
  console.log("An error occurred:", error.message);
} finally {
  console.log("Execution completed");
}
```

Output:

```
An error occurred: z is not defined
Execution completed
```

Q2. Why is error handling important in JavaScript applications?

Ans: Error handling is important in JavaScript because it helps applications manage unexpected problems without crashing and ensures a smooth user experience.

Reasons Why Error Handling Is Important

1. Prevents Application Crashes
   Without error handling, a single error can stop the entire JavaScript program. Error handling allows the application to continue running safely.
2. Improves User Experience
   Instead of showing broken pages or console errors, applications can display friendly error messages to users.
3. Easier Debugging
   Error handling helps developers identify and understand errors quickly, making debugging faster and more efficient.
4. Maintains Application Stability
   It ensures the application behaves in a controlled way even when unexpected inputs or system issues occur.
5. Handles External Failures
   Errors from APIs, network issues, or invalid user input can be managed properly using error handling.
6. Ensures Code Reliability
   Proper error handling improves the overall quality and reliability of JavaScript applications.