

Q6. What is callback hell? Explain different ways to solve callback hell with examples each.

A) Callback hell, also known as the "Pyramid of Doom," is a situation in asynchronous JavaScript programming where multiple nested callbacks are used to handle the flow of control.

Example:

```
asyncFunction1((result1) => {  
  // Do something with result1  
  asyncFunction2((result2) => {  
    // Do something with result2  
    asyncFunction3((result3) => {  
      // Do something with result3  
      // ...and so on  
    });  
  });  
});
```

Using Named Functions:

```
function handleResult1(result1) {  
  // Do something with result1  
  asyncFunction2(handleResult2);  
}
```

```
function handleResult2(result2) {  
  // Do something with result2  
  asyncFunction3(handleResult3);  
}
```

```
function handleResult3(result3) {  
  // Do something with result3  
  // ...and so on  
}
```

```
asyncFunction1(handleResult1);
```

Using Promises:

```
asyncFunction1()
  .then((result1) => {
    // Do something with result1
    return asyncFunction2();
  })
  .then((result2) => {
    // Do something with result2
    return asyncFunction3();
  })
  .then((result3) => {
    // Do something with result3
    // ...and so on
  })
  .catch((error) => {
    // Handle errors
  });
```

Using Async/Await:

```
async function myAsyncFunction() {
  try {
    const result1 = await asyncFunction1();
    // Do something with result1
    const result2 = await asyncFunction2();
    // Do something with result2
    const result3 = await asyncFunction3();
    // Do something with result3
    // ...and so on
  } catch (error) {
    // Handle errors
  }
}
```