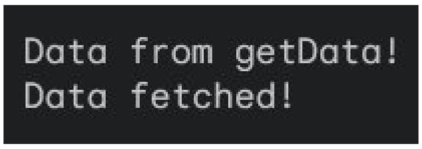
Sumedh ahire
FYMCA-B 03
BATCH 1
ASSIGNEMT 2

CODE:

```
// Approach 1: Using async/await (modern environments)
async function fetchData() {
  // Simulate asynchronous operation (e.g., network request)
  const response = await new Promise((resolve) => setTimeout(() => resolve('Data fetched!'), 1000));
  return response;
}

// Approach 2: Wrapper function (older environments or more control)
function promisify(asyncFunction) {
  return (...args) => new Promise((resolve, reject) => {
    asyncFunction(...args, (err, result) => {
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    });
  });
}

// Example asynchronous function
function getData(callback) {
  setTimeout(() => callback(null, 'Data from getData!'), 500);
}

// Convert to promise-based function using wrapper
const getDataPromise = promisify(getData);

// Usage examples
fetchData()
  .then(data => console.log(data)) // data will be 'Data fetched!' after 1 second
  .catch(error => console.error(error));

getDataPromise()
  .then(data => console.log(data)) // data will be 'Data from getData!' after 0.5 seconds
  .catch(error => console.error(error));
```

OUTPUT:



```
Data from getData!
Data fetched!
```