

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Q1. Create a user-defined function callback function that sorts an array of book prices in ascending or descending order based on the callback provided. The master function should take an array and a callback function as arguments.

Ans. function PriceSort (prices, callbackfunc) {  
    return callbackfunc(prices);  
}

function Ascendingsort (prices) {  
    prices.sort((a, b) => a-b);  
    return prices;  
}

function Descendingsort (prices) {  
    prices.sort((a, b) => b-a);  
    return prices;  
}

const ~~pro~~ bookprices = [300, 750, 280, 250, 450];

const AfterSortAsc = PriceSort(bookprices,  
    Ascendingsort);  
console.log('Ascending Order:', AfterSortAsc);

const AfterSortDesc = PriceSort(bookprices,  
    Descendingsort);  
console.log('Descending order:', AfterSortDesc);



Q2. Create an object named 'book' with at least 5 keys (eg., 'bookid', 'title', 'author', 'price', 'genres'), one of the keys should have an array as its value. Then create a function that accepts this object in a destructured format, prints all the information, and calls another function that will pass the key array into another function. Additionally, call a function that will pass the object as a key into another function and print all the keys.

Ans. 

```
const book = {  
  bookid: 'Book1',  
  title: 'Javascript Concepts',  
  author: 'Kyle Simpson',  
  price: 70.99,  
  genres: ['First 'classic', 'Conceptual', 'Guidebook']  
};
```

```
function function3(obj) {  
  Object.keys(obj).forEach(key => {  
    console.log(key);  
  });  
}
```

```
function function2(genres) {  
  console.log(" ");  
  genres.forEach(genre => {  
    console.log('genre: $genre');  
  });  
}
```



```
function printBookdetails ({ bookid, title, author,  
    price, genres }) {
```

```
    console.log('Book Id is : ${bookid}', title is :  
        `${title}`, author of book is : `${author}`,  
        book price is : `${price}`, and book  
        genres are : '');
```

```
    function2 (genres);  
    function3 (book);
```

```
}
```

```
printBookdetails (book);
```

Q4. Create an object named 'restaurant' with at least 5 keys (eg- 'restaurantid', 'name', 'cuisines', 'rating', 'reviews') ..... and print all keys.

Ans. 

```
const restaurant = {
  restaurantid: 1,
  name: 'Delight',
  cuisines: ['Italian', 'French', 'Chinese'],
  rating: 4.5,
  reviews: { user1: 'Delicious food',
             user2: 'Services are good',
             user3: 'Highly recommend' }
};
```

~~function function2(cuisines) {  
 cuisines.forEach(cuisine => {  
 console.log('~~

~~function function2(cuisines) {  
 cuisines.forEach(cuisine => {  
 console.log(`\${cuisine}`);  
 });  
}~~

~~function function3(obj) {  
 obj.forEach(reviews => {  
 console.log(`\${reviews}`);  
 });  
}~~

```
function printdetails ({restaurantid, name, cuisine,  
rating, reviews}) {
```

```
  console.log('Restaurant Id is : ${restaurantid},  
    Restaurant Name is : ${name}, Ratings  
of given to Restaurant is : ${rating}, cuisines of  
restaurant are : ');
```

```
function 2 (cuisines);
```

```
function 3 console.log('Reviews given by users  
are : ');
```

```
function 3 (reviews);
```

```
printdetails (restaurant);
```



Q5. Create an object named Library with at least 5 keys (eg. LibraryId, LibraryName, Location, books, staff). . . . . and print all the keys

Ans: `const library = {  
 LibraryId : 1,  
 LibraryName : 'All In One Library',  
 Location : 'Street 1',  
 books : {  
 book1 : 'fiction Book',  
 book2 : 'Literature Book',  
 book3 : 'Basic of Javascript' },  
 Staff : ['Aman', 'Bhavik', 'Ram'] };`

```
function function2 ( staff ) {  
  staff.forEach( staff => {  
    console.log( staff '$ { staff } ' ) ;  
  } ) ;  
}
```

```
function function3( book obj ) {  
  obj.forEach( book => {  
    console.log( 'books: $ { index } + 1 } : $ { book } ' ) ;  
  } ) ;  
}
```

```
function printLibDetails ( { LibraryId, LibraryName,  
  Location, books, Staff } ) {
```

```
console.log('Library Id is : $$LibraryId',  
            'Library name is : $$LibraryName',  
            'Location of Library is : $$Location', 'Staff of  
of Library are : ');
```

```
function2 (staff);  
console.log('books mentioned in Library are : ');  
function3 (books);
```

```
printLibdetails (library);
```



Q6. Create a function named 'addToBookCollection' that accepts an array of books and a new book to add. Each time the function is called, it should return a new array with the new book added at the end, without mutating the original array. Use the spread operator to achieve this.

Ans. 

```
function addToBookCollection(bookcollection, newbook)
{
  return [...bookcollection, newbook];
}
```

```
const initialbookcollection = [{ bookid: 'Book1' }, { bookid: 'Book2' }, { bookid: 'Book3' }];
```

```
const newbook = { bookid: 'Book4' };
```

```
const LatestCollection = addToBookCollection(initialbookcollection, newbook);
```

```
console.log('Initial book collection:');
console.log(initialbookcollection);
```

```
console.log('Updated Book collection');
console.log(LatestCollection);
```



Q7. Create a function named 'removeLastReview' that accepts an array of restaurant reviews. Each time the function is called, it should return a new array with the last review removed, without mutating the original array. Use the spread operator and array methods to achieve this.

Ans. 

```
function removeLastReview ( reviews ) {  
  return reviews.slice( 0, -1 );  
}
```

```
const restaurantReview = [ 'Nice food', 'I would  
recommend this place', 'excellent' ] ;
```

```
const RecentReviews = removeLastReview(restaurantReview) ;
```

```
console.log ( 'Earlier reviews : ' ) ;  
console.log ( restaurantReview ) ;
```

```
console.log ( 'Updated Reviews : ' ) ;  
console.log ( RecentReviews ) ;
```