# React Examples: Classes, Inheritance, and Prototypes with Props

## 1. Classes & Objects

Here, we create a basic Product class and use it in a React component.

Code Example:

import React from 'react';  
  
class Product {  
 constructor(name, price) {  
 this.name = name;  
 this.price = price;  
 }  
 getDetails() {  
 return `${this.name} costs $${this.price}`;  
 }  
}  
  
function ProductCard({ product }) {  
 return (  
 <div style={{ border: '1px solid black', margin: '10px', padding: '10px' }}>  
 <h2>Product Details:</h2>  
 <p>{product.getDetails()}</p>  
 </div>  
 );  
}  
  
function App() {  
 const product = new Product('Laptop', 1000);  
 return <ProductCard product={product} />;  
}  
  
export default App;

## 2. Inheriting Classes

Here, we extend the Product class into a DiscountedProduct class that adds a discount feature.

Code Example:

import React from 'react';  
  
class Product {  
 constructor(name, price) {  
 this.name = name;  
 this.price = price;  
 }  
 getDetails() {  
 return `${this.name} costs $${this.price}`;  
 }  
}  
  
class DiscountedProduct extends Product {  
 constructor(name, price, discount) {  
 super(name, price);  
 this.discount = discount;  
 }  
 getDiscountedPrice() {  
 return this.price - this.discount;  
 }  
}  
  
function DiscountedProductCard({ product }) {  
 return (  
 <div style={{ border: '1px solid green', margin: '10px', padding: '10px' }}>  
 <h2>Product Details with Discount:</h2>  
 <p>{product.getDetails()}</p>  
 <p>Discounted Price: ${product.getDiscountedPrice()}</p>  
 </div>  
 );  
}  
  
function App() {  
 const discountedProduct = new DiscountedProduct('Smartphone', 800, 100);  
 return <DiscountedProductCard product={discountedProduct} />;  
}  
  
export default App;

## 3. Classes & Prototypes

Here, we use prototypes to add a custom method to the Product class.

Code Example:

import React from 'react';  
  
class Product {  
 constructor(name, price) {  
 this.name = name;  
 this.price = price;  
 }  
 getDetails() {  
 return `${this.name} costs $${this.price}`;  
 }  
}  
  
// Adding a prototype method  
Product.prototype.getTaxIncludedPrice = function (taxRate) {  
 return this.price + this.price \* taxRate;  
};  
  
function ProductCardWithTax({ product }) {  
 return (  
 <div style={{ border: '1px solid blue', margin: '10px', padding: '10px' }}>  
 <h2>Product Details with Tax:</h2>  
 <p>{product.getDetails()}</p>  
 <p>Price with Tax: ${product.getTaxIncludedPrice(0.1).toFixed(2)}</p>  
 </div>  
 );  
}  
  
function App() {  
 const productWithTax = new Product('Headphones', 150);  
 return <ProductCardWithTax product={productWithTax} />;  
}  
  
export default App;