



Scala

```
/*
Code generated by ChatGPT that gives a good example what Scala could do - Chance

*/
// Define a case class for representing a product
case class Product(id: Int, name: String, price: Double)

// Define a case class for representing items in the shopping cart
case class CartItem(product: Product, quantity: Int)

// Define a class to represent the shopping cart
class ShoppingCart {
    // Initialize an empty list to store cart items
    var items: List[CartItem] = List.empty

    // Function to add a product to the cart
    def addItem(product: Product, quantity: Int): Unit = {
        if (quantity > 0) {
            val newItem = CartItem(product, quantity)
            items = newItem :: items
            println(s"Added ${quantity}x ${product.name} to the cart.")
        } else {
            println("Quantity should be greater than 0.")
        }
    }

    // Function to remove a product from the cart
    def removeItem(product: Product): Unit = {
        items = items.filterNot(_.product == product)
    }
}
```

```

        println(s"Removed ${product.name} from the cart.")
    }

// Function to calculate the total cost of items in the cart
def totalCost(): Double = {
    items.map(item => item.product.price * item.quantity).sum
}

// Function to display the contents of the cart
def displayCart(): Unit = {
    if (items.isEmpty) {
        println("The cart is empty.")
    } else {
        println("Shopping Cart Contents:")
        items.foreach(item => {
            val totalItemCost = item.product.price * item.quantity
            println(s"${item.product.name} (x${item.quantity}): $totalItemCost")
        })
        println(s"Total Cost: ${totalCost()}")
    }
}

@main def main(): Unit = {
    // Create some sample products
    val product1 = Product(1, "Laptop", 800.0)
    val product2 = Product(2, "Phone", 500.0)
    val product3 = Product(3, "Headphones", 50.0)

    // Create a shopping cart
    val cart = new ShoppingCart()

    // Add items to the cart
    cart.addItem(product1, 2)
    cart.addItem(product2, 1)
    cart.addItem(product3, 3)

    // Display the cart contents
    cart.displayCart()

    // Remove an item from the cart
    cart.removeItem(product1)

    // Display the updated cart contents
    cart.displayCart()
}

```

```
case class MainExample(id: Int, name: String, sum: Double)
@main def main(): Unit = {
    val a = BigInt(1_234_567_890_987_654_321L) // BigInt ->
    val b = BigDecimal(123_456.789)           // BigDecimal ->
    val c = 24                      // val -> immutable, kind of like final
    var msg = "Hello world! " // var -> mutable, changes over time
    msg = "It changed!!!! "
    println(msg + c)
    println(s"Kobe Bryant's number is $c") // println(s "") -> prints string
    println(s"You owe me $$c")           // do $ if you want to call variable, $$ to do literal dollar sign
}
```

- val
- var
- `println(s'')`