PROGRAM:

TEST FILE FOR TEST CASE

```
# test_ecommerce_app.py



import unittest

from ecommerce_app import ECommerceApp, Product



class ECommerceAppTestCases(unittest.TestCase):



  def setUp(self):

      # Initialize the e-commerce app instance
```

```
    self.ecommerce_app = ECommerceApp()
    self.ecommerce_app.add_product("101", "Laptop", 999.99)
    self.ecommerce_app.add_product("102", "Smartphone", 499.99)
    self.ecommerce_app.add_product("103", "Headphones", 89.99)

def test_add_product(self):
    # Test adding a new product
    result = self.ecommerce_app.add_product("104", "Tablet", 299.99)
    self.assertTrue(result, "Failed to add a new product")
    self.assertIn("104", self.ecommerce_app.products, "Product ID '104' not found in products")
```

```
def test_search_product_valid_keyword(self):
    # Test searching for a product with a valid keyword
    keyword = "laptop"
    search_results = self.ecommerce_app.search_product(keyword)
    self.assertGreater(len(search_results), 0, "No search results found for a valid keyword")



def test_search_product_invalid_keyword(self):
    # Test searching for a product with an invalid keyword
    keyword = "!@#$%"
    search_results = self.ecommerce_app.search_product(keyword)
```

```
    self.assertEqual(len(search_results), 0, "Search results found for an invalid keyword")



def test_add_to_cart(self):

    # Test adding a product to the cart

    result = self.ecommerce_app.add_to_cart("101", 1)

    self.assertTrue(result, "Failed to add product to cart")

    self.assertEqual(self.ecommerce_app.cart["101"], 1, "Cart does not have the correct quantity")



def test_checkout(self):

    # Test the checkout process

    self.ecommerce_app.add_to_cart("101", 1)
```

```
    result = self.ecommerce_app.checkout("credit_card")

    self.assertTrue(result, "Checkout process failed")

    self.assertEqual(len(self.ecommerce_app.order_history), 1, "Order history does not contain the completed order")



def test_order_history(self):

    # Test viewing order history

    self.ecommerce_app.add_to_cart("101", 1)

    self.ecommerce_app.checkout("credit_card")

    order_history = self.ecommerce_app.get_order_history()

    self.assertGreater(len(order_history), 0, "No order history found")

    self.assertEqual(order_history[0]['payment_method'], "credit_card", "Order payment method mism
```

```
atch")



if __name__ == "__main__":

  unittest.main()
```