

1. Explain how your code meets clean code practices by using at least 3 examples of your own code. Screenshots should be used
- PurchaseService is a good example of clean code as it has a single responsibility, this class only handles the checkout process

#### PurchaseService.java

```
package com.gamestore;

public class PurchaseService { 4 usages  Jonathan
    public void checkout(Cart cart) { 2 usages  Jonathan
        System.out.println("\n--- Checkout ---");
        for (Game g : cart.getGames()) {
            System.out.println(g);
        }
        System.out.printf("Total: $%.2f\n", cart.getTotal());
        cart.clear();
        System.out.println("Thank you for your purchase!");
    }
}
```

- Each class has one job
  - o Game - Stores game information
  - o Cart – Handles adding/viewing games
  - o PurchaseService – Handles buying games

#### Cart.java

```
package com.gamestore;

import java.util.ArrayList;
import java.util.List;

public class Cart { 7 usages  Jonathan
    private List<Game> games = new ArrayList<>(); 5 usages

    public void addGame(Game game) { 8 usages  Jonathan
        games.add(game);
    }

    public boolean removeGame(String name) { 3 usages  Jonathan
        return games.removeIf( Game g -> g.getName().equalsIgnoreCase(name));
    }

    public List<Game> getGames() { 7 usages  Jonathan
        return games;
    }

    public double getTotal() { 4 usages  Jonathan
        return games.stream().mapToDouble(Game::getPrice).sum();
    }

    public void clear() { 2 usages  Jonathan
        games.clear();
    }
}
```

## Game.java

```
package com.gamestore;

public class Game { 18 usages  Jonathan
    private String name; 3 usages
    private double price; 3 usages

    public Game(String name, double price) { 12 usages  Jonathan
        this.name = name;
        this.price = price;
    }

    public String getName() { return name; }

    public double getPrice() { return price; }

    @Override  Jonathan
    public String toString() { return name + " ($" + price + ")"; }
}
```

## PurchaseService.java

```
package com.gamestore;

public class PurchaseService { 4 usages  Jonathan
    public void checkout(Cart cart) { 2 usages  Jonathan
        System.out.println("\n--- Checkout ---");
        for (Game g : cart.getGames()) {
            System.out.println(g);
        }
        System.out.printf("Total: $%.2f\n", cart.getTotal());
        cart.clear();
        System.out.println("Thank you for your purchase!");
    }
}
```

- The tests are simple, easy to understand

GameTest.java

```
package com.gamestore;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class GameTest {

    @Test
    public void testGameCreation() {
        Game game = new Game( name: "Minecraft", price: 19.99);
        assertEquals( expected: "Minecraft", game.getName());
        assertEquals( expected: 19.99, game.getPrice());
    }

    @Test
    public void testToStringReturnsNameAndPrice() {
        Game game = new Game( name: "Halo", price: 29.99);
        assertEquals( expected: "Halo ($29.99)", game.toString());
    }
}
```