

## WEEK -2 Assignment

Nanda kishor Gudala

### Bug 1: Loop Output Issue (off-by-one)

```
for (let i = 0; i < customers.length; i++) {  
  console.log(customers[i].name);  
}
```

### Bug 2: filter() Not Working (missing return)

```
const activeCustomers = customers.filter((c) => c.active === true);
```

### Bug 3: Premium Increase Logic Broken (map must return + avoid mutating)

```
const updatedPremiums = customers.map((c) => {  
  if (c.age >= 50) return { ...c, premium: c.premium * 1.1 };  
  return c;  
});
```

### Bug 4: Wrong Total Premium Calculation (reduce must return)

```
const totalPremium = customers.reduce((total, c) => {  
  return total + c.premium;  
}, 0);
```

### Bug 5: Template Literal Not Printing (use backticks)

```
console.log(`Customer ${customers[0].name} has policy ${customers[0].policy}`);
```

### Bug 6: Policy Count Incorrect (use dynamic key, not .policy)

```
const policyCount = customers.reduce((count, c) => {  
  count[c.policy] = (count[c.policy] || 0) + 1;  
  return count;  
}, {});
```

### Bug 7: Risk Level Always Undefined / wrong chaining (use else-if)

```
const customersWithRisk = customers.map((c) => {  
  let riskLevel;  
  if (c.age < 35) riskLevel = "Low";  
  else if (c.age <= 50) riskLevel = "Medium";  
  else riskLevel = "High";  
});
```

```
    return { ...c, riskLevel };  
  });
```

#### **Bug 8: Active vs Inactive Count Wrong (for...in vs for...of)**

```
let active = 0, inactive = 0;
```

```
for (const c of customers) {  
  if (c.active) active++;  
  else inactive++;  
}
```

#### **Bug 9: Arrow Function Syntax Error (return / formatting)**

```
const getLifeCustomers = () =>  
  customers  
    .filter((c) => c.policy === "Life")  
    .map((c) => c.name);
```

#### **Bug 10: Sorting Mutates Original Array (copy then sort)**

```
const sortedCustomers = [...customers].sort((a, b) => b.premium - a.premium);
```