Koushik Gaddam koushikgaddam04@gmail.com

Assignment-3

Given here is a Debugging Exercise – "Find the Bug" version based on the same Insurance Customer JavaScript module we covered in class. Insurance Customer Analytics – Find the Bug'

Given Data (Correct – Do NOT Change)

```
const customers = [
  { id: 1, name: "Ravi", age: 32, policy: "Health", premium: 4800, active: true },
  { id: 2, name: "Anita", age: 51, policy: "Life", premium: 12000, active: true },
  { id: 3, name: "Kiran", age: 28, policy: "Vehicle", premium: 3500, active: false },
  { id: 4, name: "Meena", age: 45, policy: "Health", premium: 6000, active: true },
  { id: 5, name: "Suresh", age: 60, policy: "Life", premium: 18000, active: false }
];
```

Bug 1: Loop Output Issue

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

Hint: Array index & loop condition

Ans: This throws index out of range error as the variable 'i' will reach 'n' in last loop where 'n' is the length of the array. The last element will have an index 'n-1'.

Correction:

for(let i=0;i<customers.length;i++)

Bug 2: filter() Not Working

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

Hint: Return value

Ans: Since the "c.active===true" is inside the parenthesis, the function did not return anything. So we can correct in 2 ways

Correction:

1. customers.filter((c) => {return c.active})
2. customers.filter(c => c.active)

🐞 **Bug 3: Premium Increase Logic Broken**

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

🔍 **Hint:** map return & immutability

Ans: We did not return anything again.

Correction:

After the if block, add the line:

return c

🐞 **Bug 4: Wrong Total Premium Calculation**

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

🔍 **Hint:** reduce return

Ans: There is no return statement again.

Correction: Simply remove the parenthesis

🐞 **Bug 5: Template Literal Not Printing**

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

🔍 **Hint:** Quotes type

Ans: We don’t get expected result.

Correction: Use tiled quotes - ``

🐞 **Bug 6: Policy Count Incorrect**

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

🔍 **Hint:** Dynamic object key

Correction: replace count.policy with count[c.policy]

🐞 **Bug 7: Risk Level Always Undefined**

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

🔍 **Hint:** Condition chaining

Ans: The second if block overrides the first if block

Correction: Make the 2nd if block to if-else block

🐞 **Bug 8: Active vs Inactive Count Wrong**

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

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

🔍 **Hint:** for...in vs for...of

Ans: for-in loop gives the indices of customers array

Correction: Use for-of loop

🐞 **Bug 9: Arrow Function Syntax Error**

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

🔍 **Hint:** Arrow function return

Correction: Use correct implicit return formatting

🐞 **Bug 10: Sorting Mutates Original Array**

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

🔍 **Hint:** Array mutation

Ans: The original array also gets sorted

Correction: Use spread operator

sortedCustomers = [...customers].sort()