Heaven's light is our guide

Department of Electrical & Computer Engineering

LAB REPORT

Course Title
Software Engineering & Information System Design Sessional

Course Code: ECE 3118

Submitted By
Md. Nazat Kabir
Roll: 2010026
Session: 2020-21
Experiment Date: 21.04.2024
Submission Date: 15.11.2024

Submitted To
Oishi Jyoti
Assistant Professor
ECE-RUET

Experiment No. 2

Experiment Name: Report on Practicing Naming Techniques in Programming

Objectives:

Use comment to indicate the naming convention in program.

1. Write a code to manage shopping list to add, delete items to the list and display it – (Pascal)
2. Write a code to select CR where each student is represented by a structure containing ther name, roll number, marks. CR is choosen based on highest mark - (Snake + Camel)
3. Write a code to add, subtract, multiply and divide – (Use different Techinques)

Theory:

Effective naming is crucial for writing clean and understandable code. These are pre -defined sets of rules that dictate how different elements in your code, like variables, functions, and classes, should be named. They promote consistency and readability within a codebase, especially for larger projects with multiple developers. Common naming conventions include:

Camel Case: First letter lowercase, subsequent words start with uppercase (e.g., totalPrice, calculateArea).
Snake Case: All lowercase separated by underscores (e.g., total_price, calculate_area).
Pascal Case: Similar to camel case but with the first letter also uppercase (e.g., TotalPrice, CalculateArea).
Kebab Case: All lowercase separated by hyphens (e.g., total-price, calculate-area)..

Implementation:

```
// Problem 1
let ProductList = [{Id:1,ProductName:"Laptop",Quantity:20, Price:
20000},{Id:2,ProductName:"Laptop",Quantity:20, Price: 20000},
{Id:3,ProductName:"Laptop",Quantity:20, Price: 20000},
   {Id:4,ProductName:"Desktop",Quantity:10,Price:80000}];

// Update Product List Function
let UpdateProductList = (Id, ProductName, Quantity, Price) => {
         ProductList.push({Id:Id,ProductName:ProductName,Quantity:Quant ity,Price:Price});
}

let DeleteProduct = (ProductList,Id) => {
        ProductList = ProductList.filter(Product => Product.Id != Id); return ProductList;

}
// Call Update Product List
UpdateProductList(5,"Mobile",10,8000);



// Display Function
let DisplayProductList = (ProductList) =>
        { ProductList.map((Product) => { console.log(Product)
        })
}

DisplayProductList(DeleteProduct(ProductList,2));



// Problem 2
// array name in snake_naming_convention
let student_list = [
        {
                fullName:  "TAJIM  NOOR",  //  structure  property  names  in
camelCase rollNumber:
        21, marks: 89

        },
        {
        fullName: "DIBAKAR", rollNumber: 22,
        marks: 70
        },
        {
        fullName: "ADIB AL MEHMOOD",
        rollNumber: 23,
        marks: 90
        },
        {
        fullName: "SAKLAINE NOOR",
        rollNumber: 24,
        marks: 40
        },
        {
        fullName: "HAFIZ SAIMON", rollNumber:
        25,
        marks: 50
        },
        {
        fullName: "TOUFIQ ISLAM",
        rollNumber: 26,
        marks: 75
        }
];

let choose_cr = (student_list) =>
        { let highest_mark = 0; let cr_roll = 0;
        student_list.map(student => { if(highest_mark<student.marks) {
                highest_mark = student.marks; cr_roll =
                student.rollNumber;
        }
})
student_list.map(student => { if(student.rollNumber ==
        cr_roll){
        console.log(student);
        }
})
}

//call cr choose_cr(student_list);

// Problem 3

function calculate(num1, num2) { const sum =
num1 + num2;
const difference = num1 - num2; const product =
num1 * num2; const quotient = num1 / num2;
return { sum, difference, product, quotient };
}

const result = calculate(10, 5);
console.log("Camel Case Results:", result);

function calculate_numbers(number_1, number_2)
{ const total_sum = number_1 + number_2; const difference
= number_1 - number_2; const product = number_1 *
number_2; const quotient = number_1 / number_2;
return { total_sum, difference, product, quotient };
}

const result = calculate_numbers(10, 5);
console.log("Snake Case Results:", result);


function CalculateNumbers(Number1, Number2) { const Sum = Number1
+ Number2;
const Difference = Number1 - Number2; const Product =
Number1 * Number2; const Quotient = Number1 /
Number2;
return { Sum, Difference, Product, Quotient };
}

const result = CalculateNumbers(10, 5); console.log("Pascal Case
Results:", result);

function calculate-numbers(num-1, num-2)
{ const total-sum = num-1 + num-2;
const difference = num-1 - num-
2; const product = num-1 * num-
2; const quotient = num-1 / num-
2;
        return { 'total-sum': total-sum, difference, product,
quotient };
}

const result = calculate-numbers(10, 5);
console.log("Kebab Case Results:", result);
```

Discussion

Naming techniques in programming act like road signs in our code, guiding everyone to understand what things do, reducing errors, saving maintenance time, and promoting collaboration through clear and consistent communication within the code itself.

References

[1] Blog - Naming conventions in programming – a review of scientific literature: https://arxiv.org/pdf/2103.07487

[2] YouTube video - Programming Naming Conventions Every Programmer Should Know: https://www.youtube.com/watch?v=ON00cOfZhX4