| Hands-on Activity 6.1 | |
|---|---|
| **Functions** | |
| **Course Code:** CPE007 | **Program:** Computer Engineering |
| **Course Title:** Programming Logic and Design | **Date Performed: 11/23/2025** |
| **Section: CPE11S1** | **Date Submitted:11/23/2025** |
| **Name(s): Ralph Angelov F. Braganza** | **Instructor: Engr. Jimlord M. Quejado** |

**Output**

Objective:
This activity aims to develop programming skills using functions.

Intended Learning Outcomes (ILOs):
The students should be able to:

2.1 Create a program in C++ that will add, subtract, divide, multiply for 2 user-input integer values. Use functions in the program.

**ADDITION**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 1

------ Calculator ------
Enter the first number: 2.5
Enter the second number: 3
Choose operation:
1. +
2. -
3. *
4. /
Enter choice (1-4): 1
Result: 2.5 + 3 = 5.5
```

**SUBTRACTION**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 1

------ Calculator ------
Enter the first number: 2.5
Enter the second number: 3
Choose operation:
1. +
2. -
3. *
4. /
Enter choice (1-4): 2
Result: 2.5 - 3 = -0.5
```

**MULTIPLICATION**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 1

------ Calculator ------
Enter the first number: 2.5
Enter the second number: 3
Choose operation:
1. +
2. -
3. *
4. /
Enter choice (1-4): 3
Result: 2.5 * 3 = 7.5
```

**DIVISION**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 1

------ Calculator ------
Enter the first number: 2.5
Enter the second number: 3
Choose operation:
1. +
2. -
3. *
4. /
Enter choice (1-4): 4
Result: 2.5 / 3 = 0.833333
```

**DIVISION BY 0**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 1

------ Calculator ------
Enter the first number: 4
Enter the second number: 0
Choose operation:
1. +
2. -
3. *
4. /
Enter choice (1-4): 4
Denominator cannot be 0.
```

2.2 Create a program in C++ that will convert a user-input Fahrenheit to Celsius and vice versa. Use functions in the program.

**FAHRENHEIT TO CELSIUS**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 2

--- Temperature Converter ---
1. Fahrenheit to Celsius
2. Celsius to Fahrenheit
Enter your choice (1 or 2): 1
Enter temperature in Fahrenheit: 100.5
100.5 F is 38.0556 C
```

**CELSIUS TO FAHRENHEIT**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 2

--- Temperature Converter ---
1. Fahrenheit to Celsius
2. Celsius to Fahrenheit
Enter your choice (1 or 2): 2
Enter temperature in Celsius: 100.5
100.5 C is 212.9 F
```

2.3 Create a program in C++ that will convert dollars into pesos and vice versa. Use functions in the program.

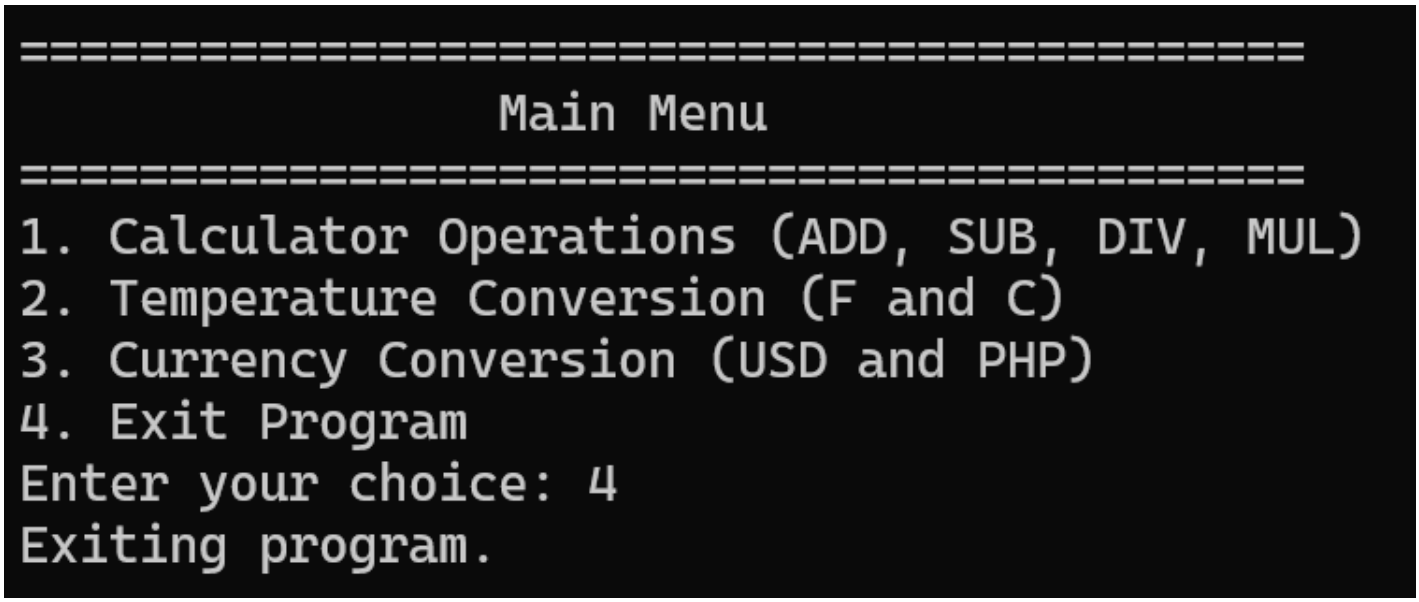
**DOLLARS TO PESOS**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 3

--- Currency Converter (Rate: 1 USD = 58.18 PHP) ---
1. Dollars (USD) to Pesos (PHP)
2. Pesos (PHP) to Dollars (USD)
Enter your choice (1 or 2): 1
Enter the amount: 100.5
$100.5 USD is 5847.09 PHP
```

**PESOS TO DOLLARS**

```
========================================
               Main Menu
========================================
1. Calculator Operations (ADD, SUB, DIV, MUL)
2. Temperature Conversion (F and C)
3. Currency Conversion (USD and PHP)
4. Exit Program
Enter your choice: 3

--- Currency Converter (Rate: 1 USD = 58.18 PHP) ---
1. Dollars (USD) to Pesos (PHP)
2. Pesos (PHP) to Dollars (USD)
Enter your choice (1 or 2): 2
Enter the amount: 6000.5
6000.5 PHP is $103.137 USD
```

**EXITING PROGRAM**




**CODE:**

```cpp
#include <iostream>

double add(double num1, double num2) {
    return num1 + num2;
}

double subtract(double num1, double num2) {
    return num1 - num2;
}

double multiply(double num1, double num2) {
    return num1 * num2;
}

double divide(double num1, double num2) {
    if (num2 == 0.0) {
        std::cout << "Denominator cannot be 0." << std::endl;
        return 0.0;
    }
    return num1 / num2;
}

const double CURRENT_RATE = 58.18;

double fahrToCels(double fahrenheit) {
    return (fahrenheit - 32.0) * (5.0 / 9.0);
}

double celsToFahr(double celsius) {
    return (celsius * (9.0 / 5.0)) + 32.0;
}

double dollarsToPesos(double amountUSD) {
    return amountUSD * CURRENT_RATE;
}

double pesosToDollars(double amountPHP) {
    return amountPHP / CURRENT_RATE;
}

void calculator_menu() {
    double num1, num2;
    int operation;

    std::cout << "\n------ Calculator ------" << std::endl;

    while (true) {
        std::cout << "Enter the first number: ";
        std::cin >> num1;
        if (!std::cin.fail()) {
            break;
        }
        std::cout << "Invalid input. Please enter a number." << std::endl;
        std::cin.clear();
        std::cin.ignore(10000, '\n');
    }

    while (true) {
        std::cout << "Enter the second number: ";
        std::cin >> num2;
        if (!std::cin.fail()) {
            break;
        }
        std::cout << "Invalid input. Please enter a number." << std::endl;
        std::cin.clear();
        std::cin.ignore(10000, '\n');
    }
```

```cpp

    while (true) {
        std::cout << "Choose operation:" << std::endl;
        std::cout << "1. +" << std::endl;
        std::cout << "2. -" << std::endl;
        std::cout << "3. *" << std::endl;
        std::cout << "4. /" << std::endl;
        std::cout << "Enter choice (1-4): ";

        std::cin >> operation;
        if (std::cin.fail()) {
            std::cout << "Invalid input. Please enter a number." << std::endl;
            std::cin.clear();
            std::cin.ignore(10000, '\n');
            continue;
        }

        switch (operation) {
            case 1:
                std::cout << "Result: " << num1 << " + " << num2 << " = " << add(num1, num2) << std::endl;
                return;
            case 2:
                std::cout << "Result: " << num1 << " - " << num2 << " = " << subtract(num1, num2) << std::endl;
                return;
            case 3:
                std::cout << "Result: " << num1 << " * " << num2 << " = " << multiply(num1, num2) << std::endl;
                return;
            case 4:
            {
                double result = divide(num1, num2);
                if (num2 != 0.0) {
                    std::cout << "Result: " << num1 << " / " << num2 << " = " << result << std::endl;
                }
                return;
            }
            default:
                std::cout << "Invalid operation selected. Please enter a number between 1 and 4." << std::endl;
        }
    }
}

void temperature_menu() {
    int choice;
    double temp;

    std::cout << "\n--- Temperature Converter ---" << std::endl;

    while (true) {
        std::cout << "1. Fahrenheit to Celsius" << std::endl;
        std::cout << "2. Celsius to Fahrenheit" << std::endl;
        std::cout << "Enter your choice (1 or 2): ";

        std::cin >> choice;
        if (std::cin.fail()) {
            std::cout << "Invalid input. Please enter a number." << std::endl;
            std::cin.clear();
            std::cin.ignore(10000, '\n');
            continue;
        }

        if (choice == 1 || choice == 2) {
            break;
        } else {
            std::cout << "Invalid choice. Please enter 1 or 2." << std::endl;
        }
    }

    while (true) {
        if (choice == 1) {
            std::cout << "Enter temperature in Fahrenheit: ";
        } else {
            std::cout << "Enter temperature in Celsius: ";
        }

        std::cin >> temp;
        if (!std::cin.fail()) {
            break;
        }
        std::cout << "Invalid input. Please enter a number for the temperature." << std::endl;
        std::cin.clear();
        std::cin.ignore(10000, '\n');
    }

    if (choice == 1) {
        std::cout << temp << " F is " << fahrToCels(temp) << " C" << std::endl;
    } else {
        std::cout << temp << " C is " << celsToFahr(temp) << " F" << std::endl;
    }
}

void currency_menu() {
    int choice;
    double amount;

    std::cout << "\n--- Currency Converter (Rate: 1 USD = " << CURRENT_RATE << " PHP) ---" << std::endl;

    while (true) {
        std::cout << "1. Dollars (USD) to Pesos (PHP)" << std::endl;
        std::cout << "2. Pesos (PHP) to Dollars (USD)" << std::endl;
        std::cout << "Enter your choice (1 or 2): ";

        std::cin >> choice;
        if (std::cin.fail()) {
            std::cout << "Invalid input. Please enter a number." << std::endl;
            std::cin.clear();
            std::cin.ignore(10000, '\n');
            continue;
        }

        if (choice == 1 || choice == 2) {
            break;
        } else {
            std::cout << "Invalid choice. Please enter 1 or 2." << std::endl;
        }
    }
```

```cpp
    while (true) {
        std::cout << "Enter the amount: ";
        std::cin >> amount;
        if (!std::cin.fail()) {
            break;
        }
        std::cout << "Invalid input. Please enter a number for the amount." << std::endl;
        std::cin.clear();
        std::cin.ignore(10000, '\n');
    }

    switch (choice) {
        case 1:
            std::cout << "$" << amount << " USD is " << dollarsToPesos(amount) << " PHP" << std::endl;
            break;
        case 2:
            std::cout << amount << " PHP is $" << pesosToDollars(amount) << " USD" << std::endl;
            break;
        default:
            break;
    }
}

int main() {
    int choice = 0;

    do {
        std::cout << "\n=======================================" << std::endl;
        std::cout << "              Main Menu" << std::endl;
        std::cout << "=======================================" << std::endl;
        std::cout << "1. Calculator Operations (ADD, SUB, DIV, MUL)" << std::endl;
        std::cout << "2. Temperature Conversion (F and C)" << std::endl;
        std::cout << "3. Currency Conversion (USD and PHP)" << std::endl;
        std::cout << "4. Exit Program" << std::endl;
        std::cout << "Enter your choice: ";

        std::cin >> choice;
        if (std::cin.fail()) {
            std::cout << "Invalid input. Please enter a number." << std::endl;
            std::cin.clear();
            std::cin.ignore(10000, '\n');
            continue;
        }

        switch (choice) {
            case 1:
                calculator_menu();
                break;
            case 2:
                temperature_menu();
                break;
            case 3:
                currency_menu();
                break;
            case 4:
                std::cout << "Exiting program." << std::endl;
                break;
            default:
                std::cout << "Invalid choice. Please enter a number between 1 and 4." << std::endl;
                break;
        }

    } while (choice != 4);

    return 0;
}
```

## Supplementary Activity

**CODE ANALYSIS:**

The C++ program I've created performs basic calculator operations, temperature conversions, and currency conversion that is all in one program instead of making a dedicated program for each conversion and calculation.

For the Calculator functions we have defined four functions which are the add, subtract, multiply and divide; which will be able to perform the basic calculating operations on two numbers. The divide performs division but I've also included a check to prevent it from dividing by zero. This will only print the message "Denominator cannot be 0." if the second number is a zero and returns 0.0 to avoid a runtime error (the function is returning a floating-point value that is why it's 0.0 and not 0).

The next function is for the Conversions which handles the conversion of temperature (Fahrenheit to Celsius vice versa) and the currency (USD to Peso vice versa) by applying the formulas.The temperature conversion functions, fahrToCels(double fahrenheit) and celsToFahr(double celsius), allow the user to easily switch between the Fahrenheit and Celsius scales using standard scientific equations:

```
double fahrToCels(double fahrenheit) {
    return (fahrenheit - 32.0) * (5.0 / 9.0);
}

double celsToFahr(double celsius) {
```

```
    return (celsius * (9.0 / 5.0)) + 32.0;
}
```

Meanwhile, the currency conversion functions

```
double dollarsToPesos(double amountUSD) {
    return amountUSD * CURRENT_RATE;
}

double pesosToDollars(double amountPHP) {
    return amountPHP / CURRENT_RATE;
}
```

Will allow the user to convert between U.S. dollars (USD) and Philippine pesos (PHP). These conversions are based on a constant exchange rate defined in the program as CURRENT_RATE = 58.18 (this is the current rate as of now), meaning one U.S. dollar is equivalent to 58.18 pesos. The first function multiplies the dollar amount by this rate to get the peso value, while the second divides the peso amount by the same rate to determine its dollar equivalent.

Next we have the menus for each of the following tasks, the calculator_menu() function handles user interaction for basic calculations (add, subtract, multiply and divide). It will first ask the user to enter two numbers but it will check if the input is valid (not letters or symbols) using the std::cin.fail() function. If an invalid input was detected, it clears the error and asks again. Once both of the two numbers are inputted it will then present a list of operations: addition, subtraction, multiplication, and division. Once the user chooses an operation (1–4), the program performs the calculation using the corresponding function and displays the result. The temperature_menu() function allows the user to convert temperatures between Fahrenheit and Celsius. This function will ask whether the user wants to convert from Fahrenheit to Celsius (option 1) or Celsius to Fahrenheit (option 2). It then takes input from the user using std::cin >> choice; and immediately after that, it checks if the input is valid using the condition if (std::cin.fail()), If std::cin.fail() returns true, it means the user entered something invalid (like a letter, symbol, or blank input instead of a number). So when a invalid input happens it will print an error message  "Invalid input. Please enter a number." then calls std::cin.clear(); to clear the error flags set on the stream. Uses std::cin.ignore(10000, '\n'); to remove any leftover characters from the input and the loop repeats, asking the user to input again. If the input is valid but not within the correct range (for example, not 1 or 2), it will also print another message "Invalid choice. Please enter 1 or 2." (this will do the same for all the other functions). Once a valid input is received, it calls the appropriate conversion function and displays the converted value with the correct units.

The currency_menu() function converts money between USD and PHP. It displays the current exchange rate (1 USD = 58.18 PHP) then the user chooses whether they want to convert from dollars to pesos or vice versa. Once the user enters a valid conversion choice, it will proceed to ask for the amount of money to convert and once both the conversion type and amount are valid, it will then use a switch statement to perform the correct operation.

```
switch (choice) {
    case 1:
        std::cout << "$" << amount << " USD is " << dollarsToPesos(amount) << " PHP" << std::endl;
        break;
    case 2:
        std::cout << amount << " PHP is $" << pesosToDollars(amount) << " USD" << std::endl;
        break;
    default:
        break;
}
```

Each result is displayed clearly, showing both the original amount and the converted amount, along with their currency symbols.

Here we have the main() function which is the core of the program. It displays the main menu options and controls which section of the program to execute based on the user's input. Inside main(), a variable is declared to store the user's choice int choice = 0; then the main function is inside a do-while loop, which keeps running until the user chooses to exit. The program keeps looping  until the user selects option 4 which exits the program and every time the loop begins, the program prints the Main Menu to clearly list all available functions for the user. After displaying the menu, the program will ask for an input from the user but before it proceeds, it checks for input errors using the block of code (this was mentioned before) and once an input has been entered the program uses a switch statement to determine what to do.

```
switch (choice) {
   case 1:
      calculator_menu();
      break;
   case 2:
      temperature_menu();
      break;
   case 3:
      currency_menu();
      break;
   case 4:
      std::cout << "Exiting program." << std::endl;
      break;
   default:
      std::cout << "Invalid choice. Please enter a number between 1 and 4." << std::endl;
      break;
}
```

If the user types 1, the function calculator_menu() is called and this function opens the sub-menu that lets the user perform addition, subtraction, multiplication, or division with two numbers. If the user chooses 2, the program calls temperature_menu(), where they can convert between Fahrenheit and Celsius. And if the user picks 3, the function currency_menu() runs. It converts money between U.S. dollars and Philippine pesos using the fixed exchange rate (the exchange rate may change though in real life). Finally if the user selects 4, the program prints the message then the loop condition choice != 4 becomes false, so the program ends but if the user enters any other number (like 0, 5, or 69), the program prints an error message "Invalid choice. Please enter a number between 1 and 4." then proceeds to loop back to show the menu again.

**Conclusion**

This hands-on activity was yet another challenging task that I've gone through, because there were so many things that just didn't work (especially the functions). Through this activity, however, it has improved my knowledge about functions, and it may not be a significant boost to my knowledge because I know for a fact I might forget most of it in the future due to other subjects piling up, but it gave me a roadmap. Even though the process was frustrating at times, I realized that every error I encountered helped me understand how functions really work in C++, from how parameters are passed to how values are returned and reused in different parts of the program. Though I might have lost a couple of brain cells here and there due to me calling the functions from the wrong places in the code, it gave me an assurance that I am learning through my mistakes as I go on. Writing and testing each section helped me see the logic behind conditionals, loops, and switch statements, and even though I've searched up and crammed most of the things I've coded so far, it has expanded my knowledge at least a little bit. I wish I could immerse myself fully in coding even more, but due to my limited time, like with other subjects, my personal life in the house, and my relationship, I just really don't have time to give myself that mastery on each topic that I learn because, and I'm going to be honest, I haven't even mastered the basics yet from our first module. Nonetheless, I did okay even with the lack of knowledge I have, but I hope and pray that sometime in the future I can fully immerse myself into coding and make original work of my own from my skills.