Functions in C - Module - 3

Function Arguments and Return Value

Function Categories

There are four main categories of the functions these are as follows:
1. Function with no arguments and no return values.
2. Function with no arguments and a return value.
3. Function with arguments and no return values.
4. Function with arguments and return values.

Type 1: Function with no arguments and no return values

- Definition: This type of function doesn't take any input (arguments) and doesn't return any output.
- Use Case: Useful for tasks that simply perform an operation without needing data or returning results.

Example:

```c
#include <stdio.h>

void greet() {  // No arguments, no return value
    printf("Hello, World!\n");
}

int main() {
    greet();  // Call the function
    return 0;
}
```

- Explanation: `greet()` prints a message and doesn't require or provide additional data.

Type 2: Function with no arguments and a return value

- **Definition**: This type of function doesn't take any input but returns a result.
- **Use Case**: Useful for generating or calculating values that don't depend on external inputs.

Example:

```c
#include <stdio.h>

int getNumber() {  // No arguments, returns an integer
    return 42;
}

int main() {
    int number = getNumber();  // Store the returned value
    printf("The number is: %d\n", number);
    return 0;
}
```

- Explanation: `getNumber()` always returns `42` without needing input.

Type 3: Function with arguments and no return value

- **Definition**: This type of function accepts input (arguments) but doesn't return any result.
- **Use Case**: Useful for performing operations based on provided data without needing to return a result.

Example:

```c
#include <stdio.h>

void printSum(int a, int b) {  // Takes two arguments, no return value
    printf("The sum is: %d\n", a + b);
}

int main() {
    printSum(3, 7);  // Call the function with arguments
    return 0;
}
```

- Explanation: `printSum()` calculates and prints the sum but doesn't return the sum to the caller.

Type 4: Function with arguments and return values

- **Definition:** This type of function accepts input and returns a result.
- **Use Case:** Useful for processing data and providing the result for further use.

Example:

```c
#include <stdio.h>

int multiply(int x, int y) {  // Takes two arguments and returns their product
    return x * y;
}

int main() {
    int result = multiply(5, 6);  // Call the function and store the return value
    printf("The product is: %d\n", result);
    return 0;
}
```

- Explanation: `multiply()` calculates and returns the product of `x` and `y`.

Solved Example: Find Maximum of Two numbers using Functions

```c
#include <stdio.h>

// Function to find maximum
int max(int x, int y) {
    if (x > y)
        return x;
    else
        return y;
}

int main() {
    int a = 25, b = 40;
    int maximum = max(a, b);  // Function call
    printf("The maximum of %d and %d is %d\n", a, b, maximum);
    return 0;
}
```

Assignment

Ques 1: Power Function

- Write a C program that calculates the power of a number.
- The program should accept a base and an exponent from the user and calculate the result using a function.
- You can use a loop to multiply the base with itself for the exponent number of times.

Ques 2: Write a program to calculate the area and perimeter of a trapezoid. The user will input the lengths of the two parallel sides and the height.
Formula:

- Area = 0.5 × (base1 + base2) × height
- Perimeter = base1 + base2 + side1 + side2 (side lengths are also required)

Ques 3: Write a program to calculate the area and perimeter (circumference) of a circle. The user will input the radius of the circle.
Formula:

- Area = $\pi \times radius^2$
- Perimeter (Circumference) = $2 \times \pi \times radius$