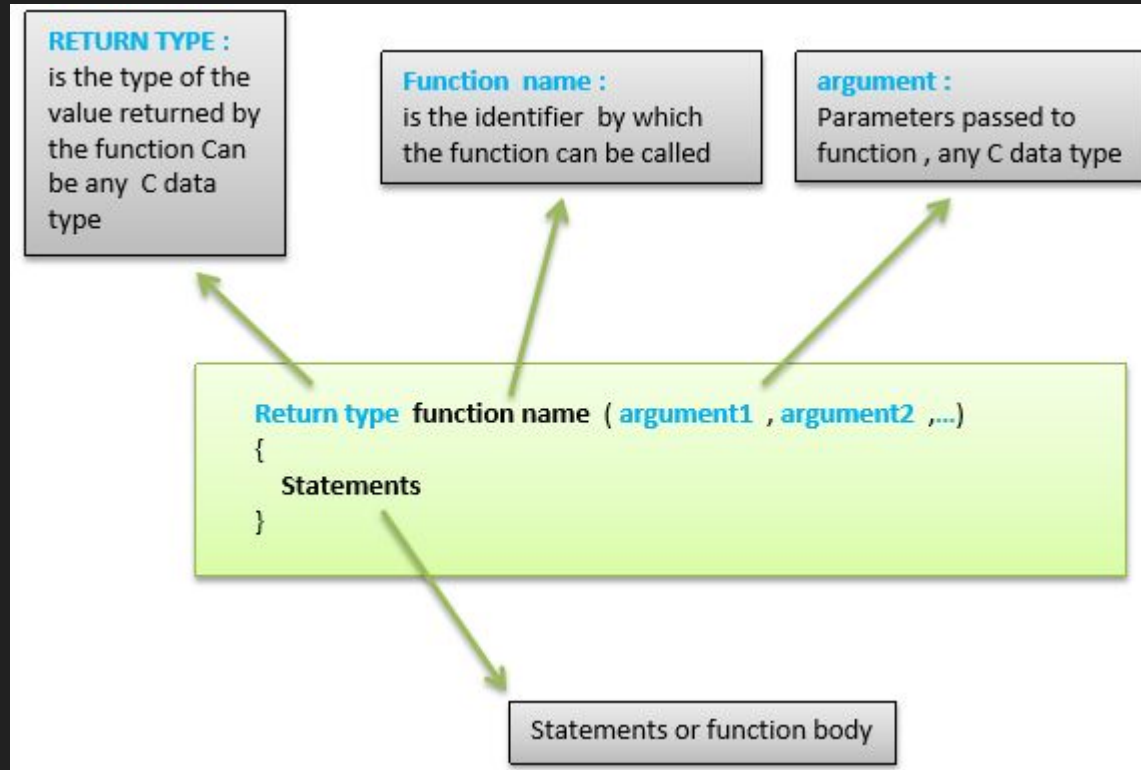


# Arduino C++ Programming

Advanced Concepts

# Functions



# Function Declaration

```
int sum_func (int x, int y) // function declaration {  
    int z = 0;  
    z = x+y ;  
    return z; // return the value  
}
```

```
void setup () {  
    Statements // group of statements  
}
```

```
Void loop () {  
    int result = 0 ;  
    result = Sum_func (5,6) ; // function call  
}
```

# Anatomy of a Function

## Anatomy of a C function

Datatype of data returned,  
any C datatype.

"void" if nothing is returned.

Parameters passed to  
function, any C datatype.

Function name

```
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Return statement,  
datatype matches  
declaration.

Curly braces required.

```
void loop(){  
    int i = 2;  
    int j = 3;  
    int k;
```

```
    k = myMultiplyFunction(i, j); // k now contains 6  
}
```

# Another Example

```
void setup() {  
    Serial.begin(9600);  
  
    DashedLine();  
    Serial.println("| Program Menu |");  
    DashedLine();  
}  
  
void loop() {  
}  
  
void DashedLine()  
{  
    Serial.println("-----");  
}
```

# Analysis

```
void setup() {  
    Serial.begin(9600);  
  
    DashedLine(); ← Function is called here  
    Serial.println("| Program Menu |");  
    DashedLine(); ← Function is called again  
}  
  
void loop() {  
}  
  
void DashedLine()  
{  
    Serial.println("-----");  
}
```

} Function is created here

# Questions and Exercises

1. Write and test the following code

```
void setup() {  
    float area;  
  
    Serial.begin(9600);  
    // calculate the area of a circle with radius of 9.2  
    area = CircleArea(9.2);  
    Serial.print("Area of circle is: ");  
    // print area to 4 decimal places  
    Serial.println(area, 4);  
}  
  
void loop() {  
}  
  
// calculate the area of a circle  
float CircleArea(float radius)  
{  
    float result;  
  
    result = 3.141592654 * radius * radius;  
  
    return result;  
}
```

# Questions and Exercises

2. Write and test a function that takes 3 numbers and returns their average.
3. Write a function that takes a number and prints out its multiplication table. The function does not return anything and so its declaration should begin with void.
4. Write a function called max that takes two ints and returns the larger of the two.
5. Write a function called min that takes two ints and returns the smaller of the two.
6. Write a function called asterisks that takes an int and displays a line of asterisks equivalent in size to the int passed to the function.



## 2. Write and test a function that takes 3 numbers and returns their average.

```
// Function to calculate average of 3 numbers
float averageOfThree(float a, float b, float c) {
    return (a + b + c) / 3.0;
}
```

```
void setup() {
    Serial.begin(9600);
```

```
    // Test the function
    float avg = averageOfThree(10, 20, 30);
```

```
    Serial.print("Average: ");
    Serial.println(avg, 2); // 2 decimal places
}
```

3. Write a function that takes a number and prints out its multiplication table. The function does not return anything and so its declaration should begin with void.

```
// Function to print multiplication table of a number
void printMultiplicationTable(int number) {
    Serial.print("Multiplication table for ");
    Serial.println(number);

    for (int i = 1; i <= 10; i++) {
        Serial.print(number);
        Serial.print(" x ");
        Serial.print(i);
        Serial.print(" = ");
        Serial.println(number * i);
    }
}

void setup() {
    Serial.begin(9600);
    delay(500); // Allow Serial Monitor to start

    printMultiplicationTable(7); // Test with number 7
}
```

## 4. Write a function called max that takes two ints and returns the larger of the two.

```
// Function that returns the larger of two integers
```

```
int max(int a, int b) {  
    if (a > b) {  
        return a;  
    } else {  
        return b;  
    }  
}
```

```
void setup() {  
    Serial.begin(9600);
```

```
    int num1 = 42;  
    int num2 = 78;
```

```
    int larger = max(num1, num2);
```

```
    Serial.print("Larger number between ");  
    Serial.print(num1);  
    Serial.print(" and ");  
    Serial.print(num2);  
    Serial.print(" is: ");  
    Serial.println(larger);  
}
```

## 5. Write a function called min that takes two ints and returns the smaller of the two.

// Function that returns the smaller of two integers

```
int min(int a, int b) {  
    if (a < b) {  
        return a;  
    } else {  
        return b;  
    }  
}
```

```
void setup() {  
    Serial.begin(9600);
```

```
    int num1 = 25;  
    int num2 = 40;
```

```
    int smaller = min(num1, num2);
```

```
    Serial.print("Smaller number between ");  
    Serial.print(num1);  
    Serial.print(" and ");  
    Serial.print(num2);  
    Serial.print(" is: ");  
    Serial.println(smaller);  
}
```

6. Write a function called `asterisks` that takes an `int` and displays a line of asterisks equivalent in size to the `int` passed to the function.

```
void asterisks(int count) {  
    for (int i = 0; i < count; i++) {  
        Serial.print("*");  
    }  
    Serial.println(); // Move to the next line after printing  
}
```

```
void setup() {  
    Serial.begin(9600);  
    delay(500); // Small delay to let Serial Monitor start
```

```
    int numStars = 10;  
    Serial.print("Printing ");  
    Serial.print(numStars);  
    Serial.println(" asterisks:");
```

```
    asterisks(numStars); // Call the function  
}
```