SOFTWARE ENGINEERING SESSION 1

Presented by

Garmin Software Engineers

https://tinyurl.com/engdaycamp

G-ROVER: SOFTWARE GOALS

1. Move the Car with Motors
2. Control the Car with Joystick
3. (Challenge) Control the Brightness & Color of LEDs
4. (Challenge) Move the Servo Motor Arm
5. (Challenge) Transmit & Receive Messages using IR

SOFTWARE SESSIONS

Today (1:15pm-2pm)

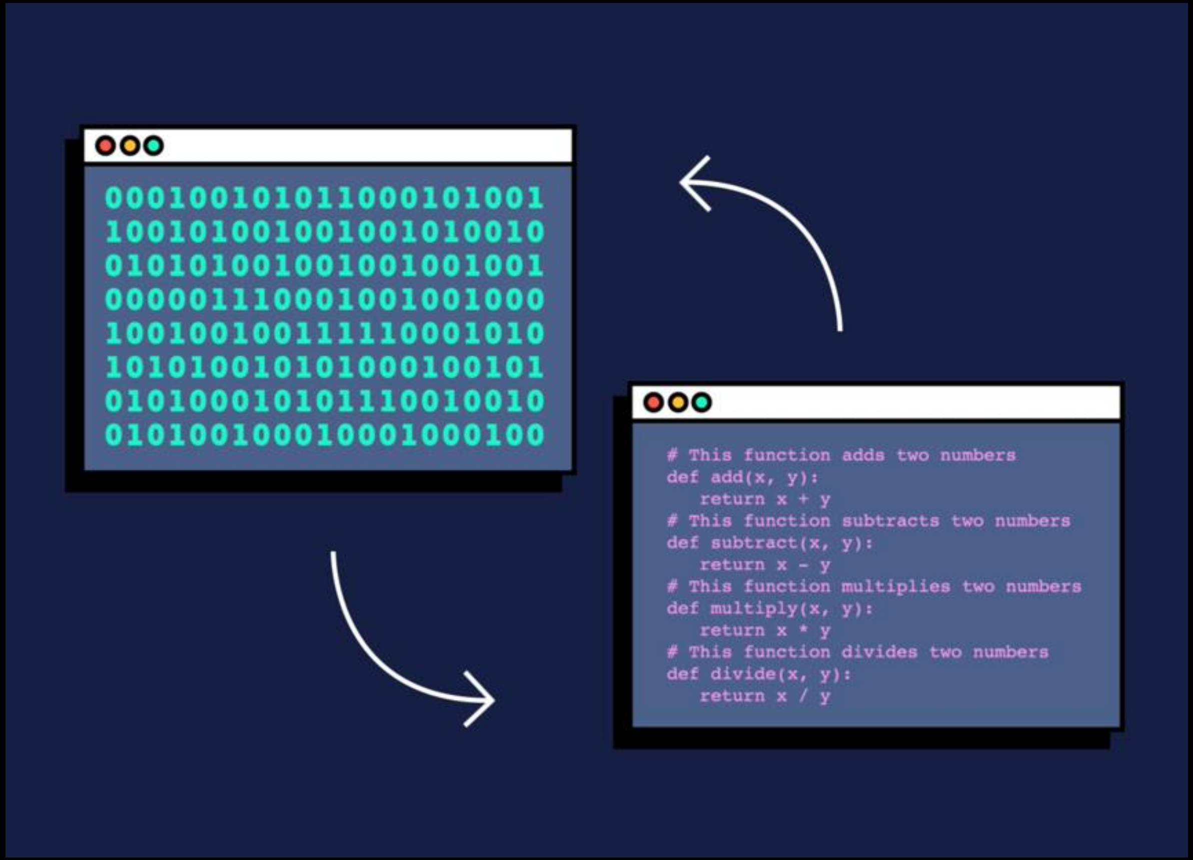
- Programming Introduction
- C++ Coding Background
- Setup Your environment

Tomorrow (12:30pm-2pm)

- Programming Your Own Car Code
- Challenge Sessions
  - Satellite Uplink: IR Communication
  - Probe Button Jumpstart: Servo Motor

WHAT IS PROGRAMMING?

- Giving instructions to a computer
- A computer reads lines of code step by step
- Different programming languages are just different ways to give the instructions
- Ex: Python, C, C++, Java, etc.



```
000100101011000101001
100101001001001010010
010101001001001001001
000001110001001001000
100100100111110001010
101010010101000100101
010100010101110010010
010100100010001000100
```

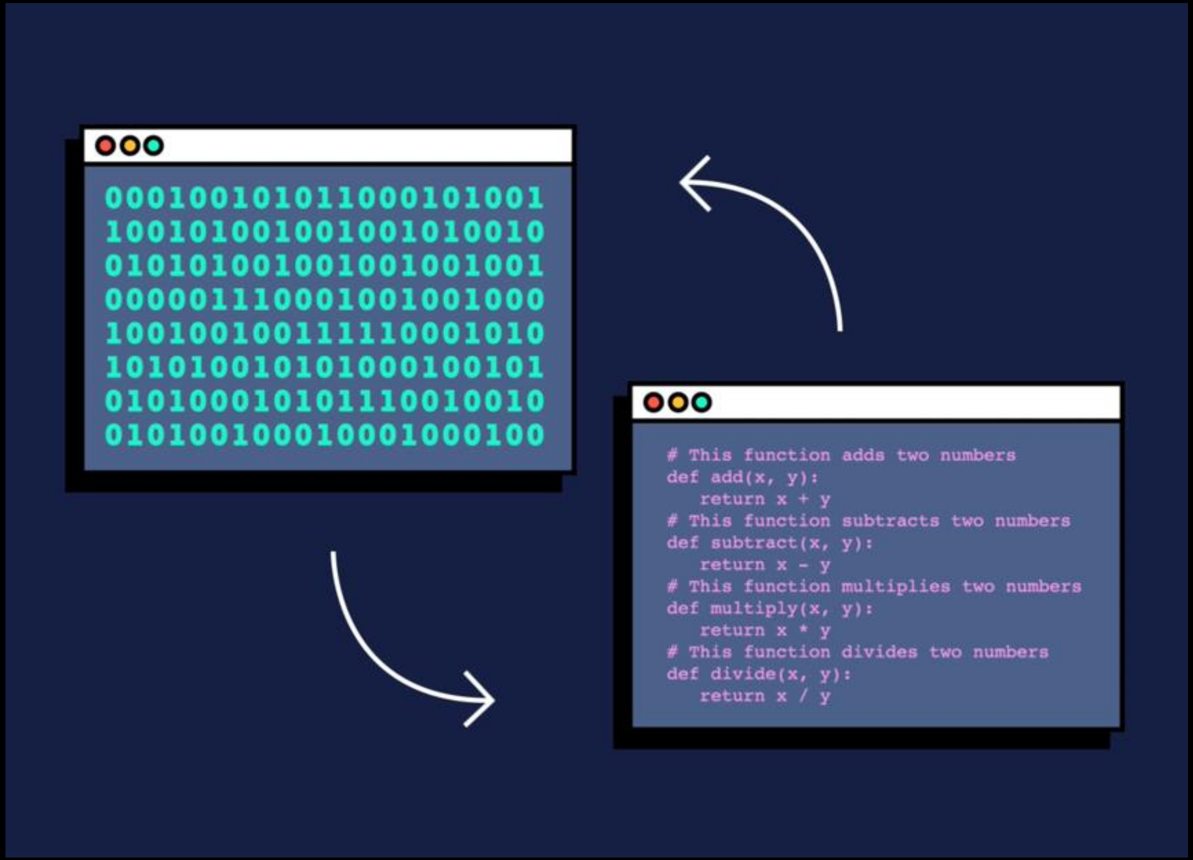
```
# This function adds two numbers
def add(x, y):
   return x + y
# This function subtracts two numbers
def subtract(x, y):
   return x - y
# This function multiplies two numbers
def multiply(x, y):
   return x * y
# This function divides two numbers
def divide(x, y):
   return x / y
```

PROGRAMMING LANGUAGE SYNTAX

- "Syntax" = the rules on how programmers are allowed to write code in a certain language for the computer to understand
- Ex: (Python) print "Hello World"

  (C++) printf("Hello World");
- In C++, never forget the semicolon!

  ;



```
0001001010110001010 01
```

```
0001001010101100010 1001
```

```
000100101010110001 01001
1001010010010010 10010
```

```
000100101010110001010 01
```

```
00010010101011000101001
10010100100100101010010
01010100100100100100 1
```

```
0001001010101100010100 1
```

```
00010010101011000101001
10010100100100101010010
01010100100100100101001
00000111000100100100 0
```

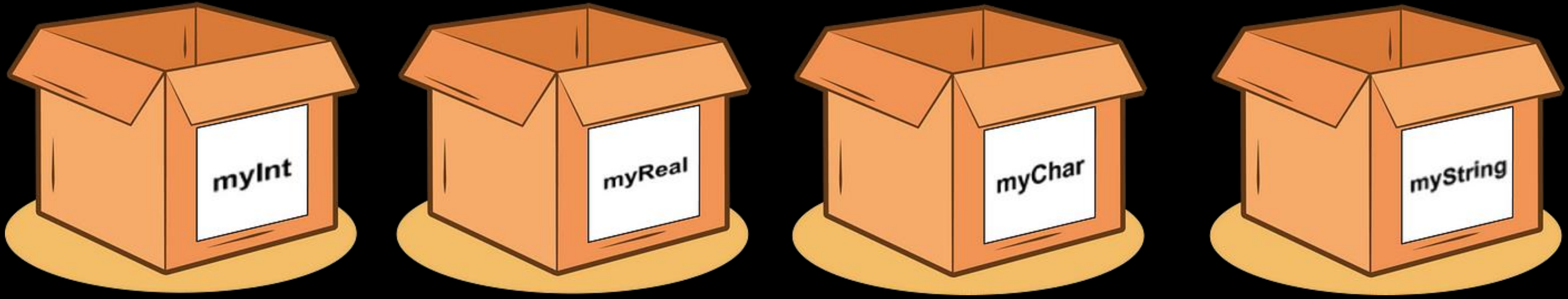
```
00010010101011000101001
10010100100100101010010
01010100100100100101001
00000111000100100101000
10010010011111100010100
10101001010101000100101
01010001010111001001 0
01010010001000100010 0
```

```
# This function adds two numbers
def add(x, y):
   return x + y
# This function subtracts two numbers
def subtract(x, y):
   return x - y
# This function multiplies two numbers
def multiply(x, y):
   return x * y
# This function divides two numbers
def divide(x, y):
   return x / y
```

C++

- A very popular programming language
- "Object-oriented" = we can store information/data in unique ways that are easy to use
- 3 main topics to learn:
  1. Variables
  2. Functions
  3. Classes

VARIABLES



- Containers for data values
- Make sure your variable names are descriptive, so you know what they are later!

VARIABLE TYPES

Variable Type	C++ Keyword	Value Ranges
Boolean	bool	True (1), False (0)
Integer	int	. . ., -2, -1, 0, 1, 2, . . .
Character	char	A, B, . . ., a, b, . . ., !, *, . . .
String	char*, string	"hello", ". . . . .!",

- Syntax: *type variableName = value;*
- Variable Example

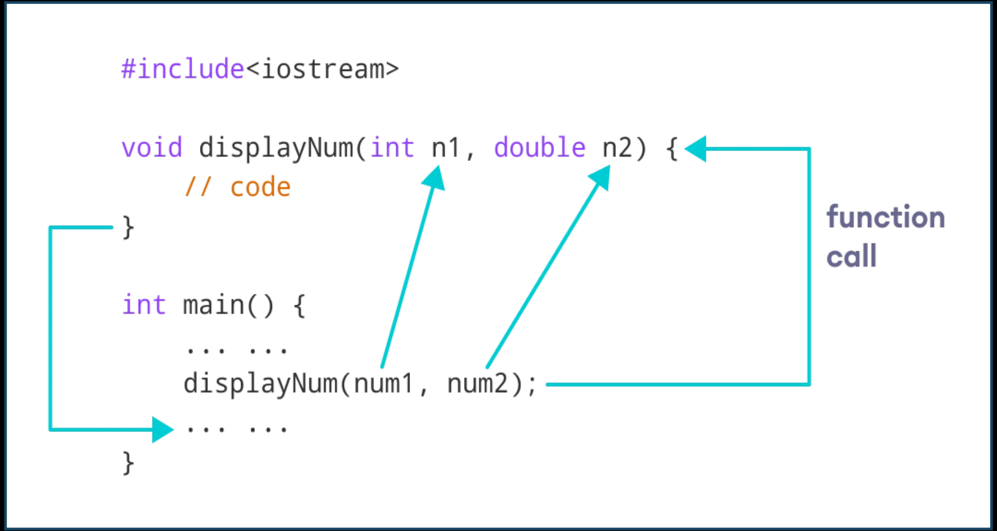
FUNCTIONS

- Functions perform a specific set of instructions
- Good for code that will be reused a lot
  - Drive Function Example
- The computer will read all the lines in a function before continuing in the main program
- Function Example 1

FUNCTIONS

```
returnType functionName(parameterType parameterName, ...) {
    // code
    return valueToReturn;
}
```

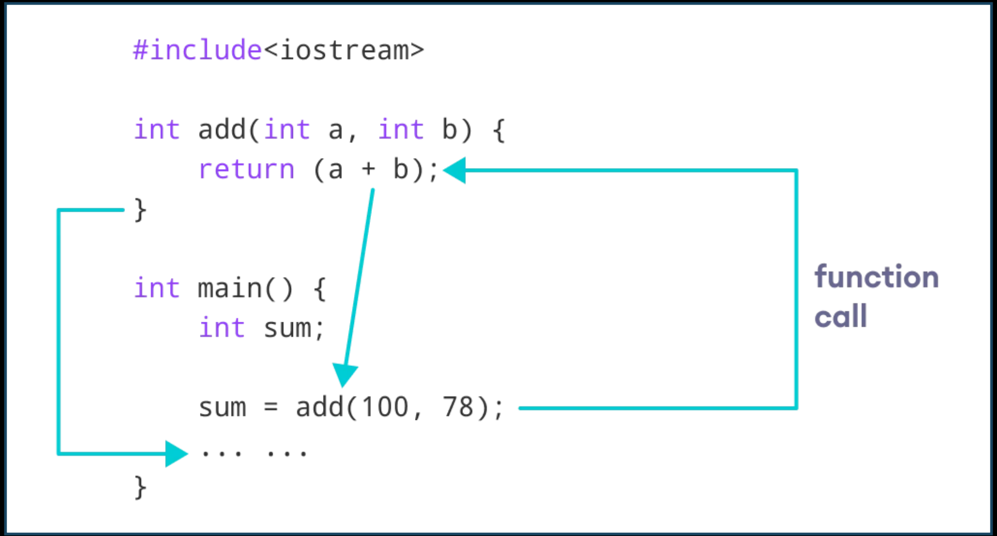
Function Example 2



```
#include<iostream>

void displayNum(int n1, double n2) {
    // code
}

int main() {
    ... ...
    displayNum(num1, num2);
    ... ...
}
```



```
#include<iostream>

int add(int a, int b) {
    return (a + b);
}

int main() {
    int sum;

    sum = add(100, 78);
    ... ...
}
```

CLASSES

- Classes are like blueprints for different data objects
- They hold variables and functions that are specific to that class – these are called "members"

```
class Circle {
public:
    // Attributes
    double radius;

    // Methods
    double calculateArea() {
        return 3.14159 * radius * radius;
    }

    double calculateCircumference() {
        return 2 * 3.14159 * radius;
    }
};
```

CLASSES

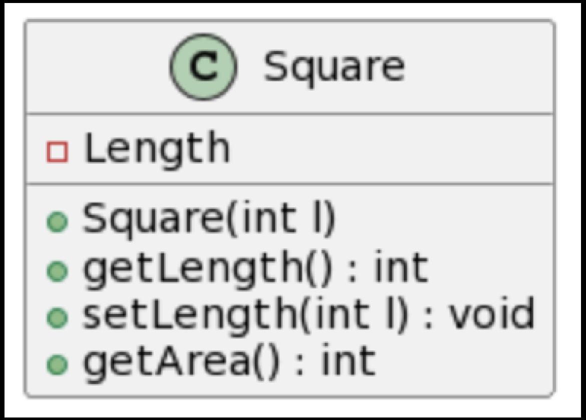
- To use a class, you must create an object of the class type and then refer to the members
- You can see the members of a class in VSCode using a "." or "->"
- Class Example

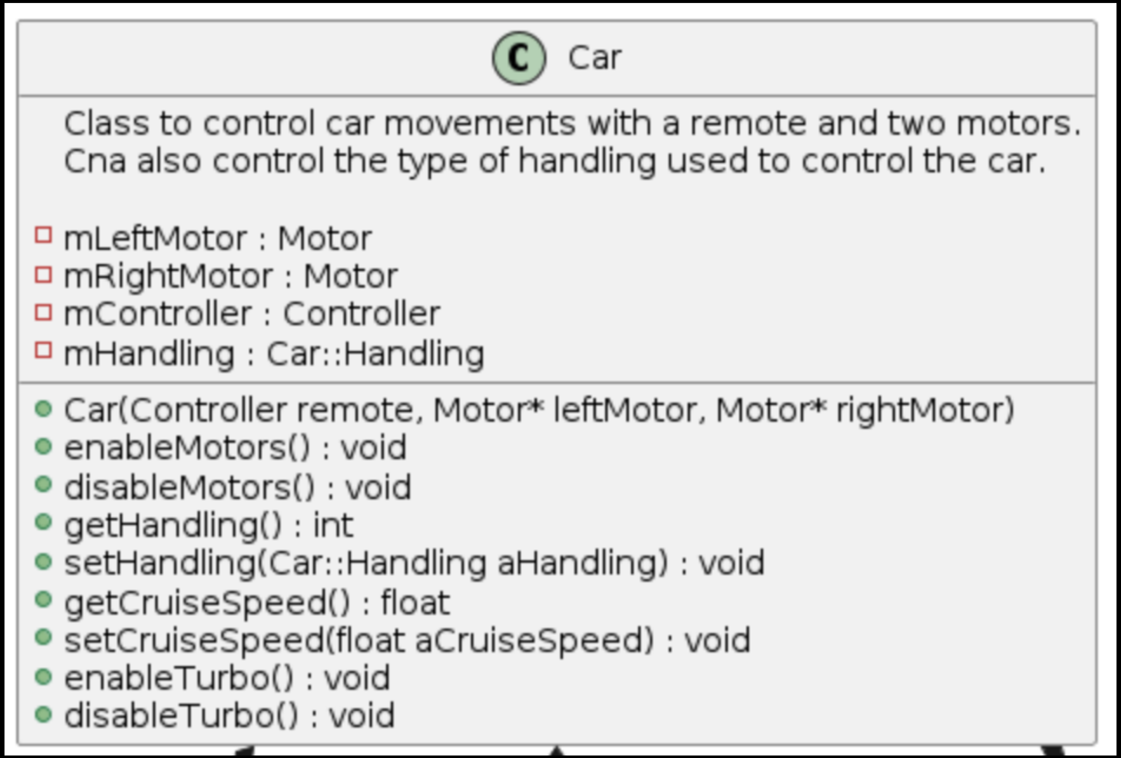
```
Circle myCircle; // Creating an instance of Circle

myCircle.radius = 5.0; // Setting the radius attribute

double area = myCircle.calculateArea();
double circumference = myCircle.calculateCircumference();
```

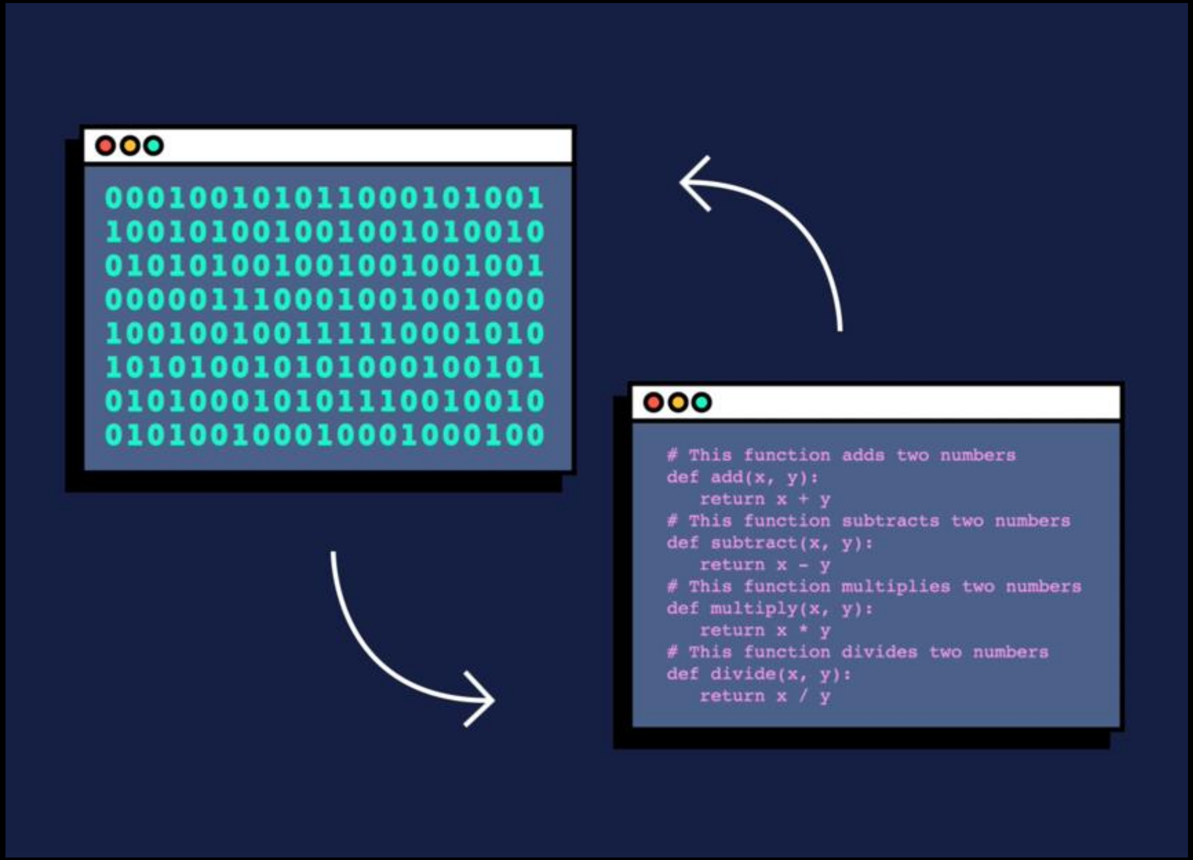
CLASSES


Square
Length
Square(int l)
getLength() : int
setLength(int l) : void
getArea() : int


Car
Class to control car movements with a remote and two motors.
Cna also control the type of handling used to control the car.
mLeftMotor : Motor
mRightMotor : Motor
mController : Controller
mHandling : Car::Handling
Car(Controller remote, Motor* leftMotor, Motor* rightMotor)
enableMotors() : void
disableMotors() : void
getHandling() : int
setHandling(Car::Handling aHandling) : void
getCruiseSpeed() : float
setCruiseSpeed(float aCruiseSpeed) : void
enableTurbo() : void
disableTurbo() : void

WHAT HAPPENS WHEN SOMETHING GOES WRONG?

- We call solving issues in code "debugging"
- Different Ways to Debug:
  - Printing out information as the program runs
  - Lights/other signals when disconnected from a computer
  - Reread and ask for help



```
0001001010110001010001
1001010010010010100010
0101010010010010010001
0000011100010010010000
1001001001111100010100
1010100101010001001011
0101000101011100100100
0101001000100010000100
```

```
# This function adds two numbers
def add(x, y):
   return x + y
# This function subtracts two numbers
def subtract(x, y):
   return x - y
# This function multiplies two numbers
def multiply(x, y):
   return x * y
# This function divides two numbers
def divide(x, y):
   return x / y
```

SEE YOU TOMORROW!