UNLV | Howard R. Hughes College of ENGINEERING

DASL-100.2 C++ Programming and Linux

DASL Drones and Autonomous Systems Lab @ UNLV

Week 3-1

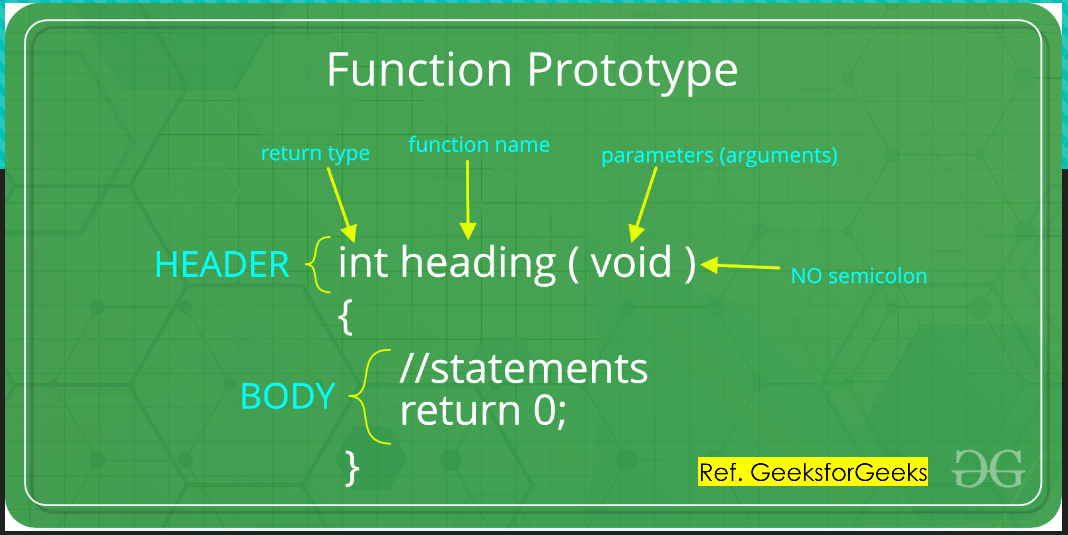
1. Function
2. Structure
3. Class

Instructor: Truc Tran

02-11-2023

1. Function

- In C++, function is a named block of code that can perform a specific task.
- Functions provide a way to modularize a program and make it easier to understand, maintain, and reuse.
- Functions can accept input in the form of arguments and can return a value back to the calling code.
- A function in C++ can return or take as input any type of data, including:
  - Built-in data types such as **int**, **float**, **double**, **char**, **bool**, etc.
  - C++ Standard Library data types, such as **string**,
  - **void** if the function does not return a value.



Function Prototype

```
int heading ( void )
{
    //statements
    return 0;
}
```

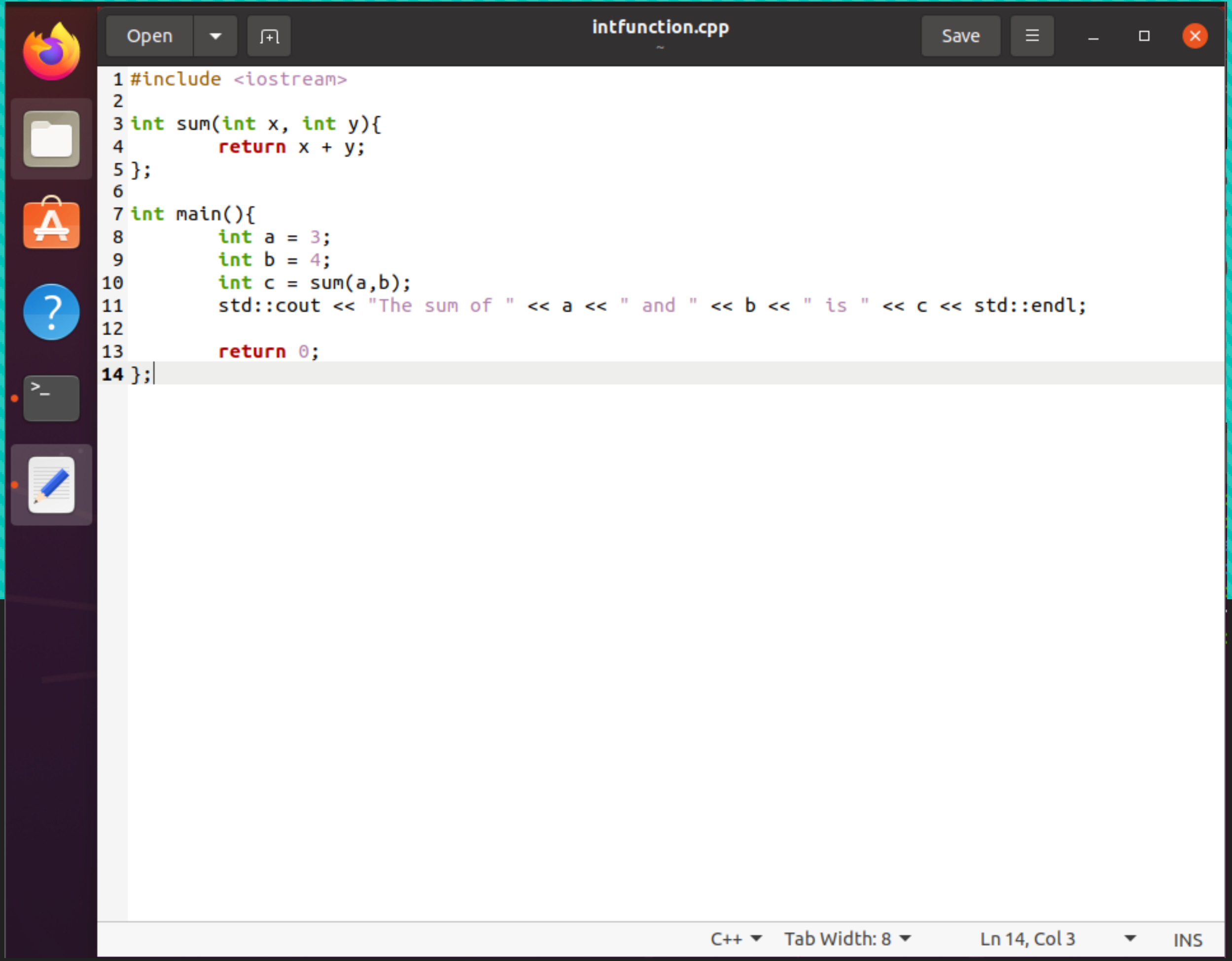
Ref. GeeksforGeeks

UNLV Howard R. Hughes College of Engineering

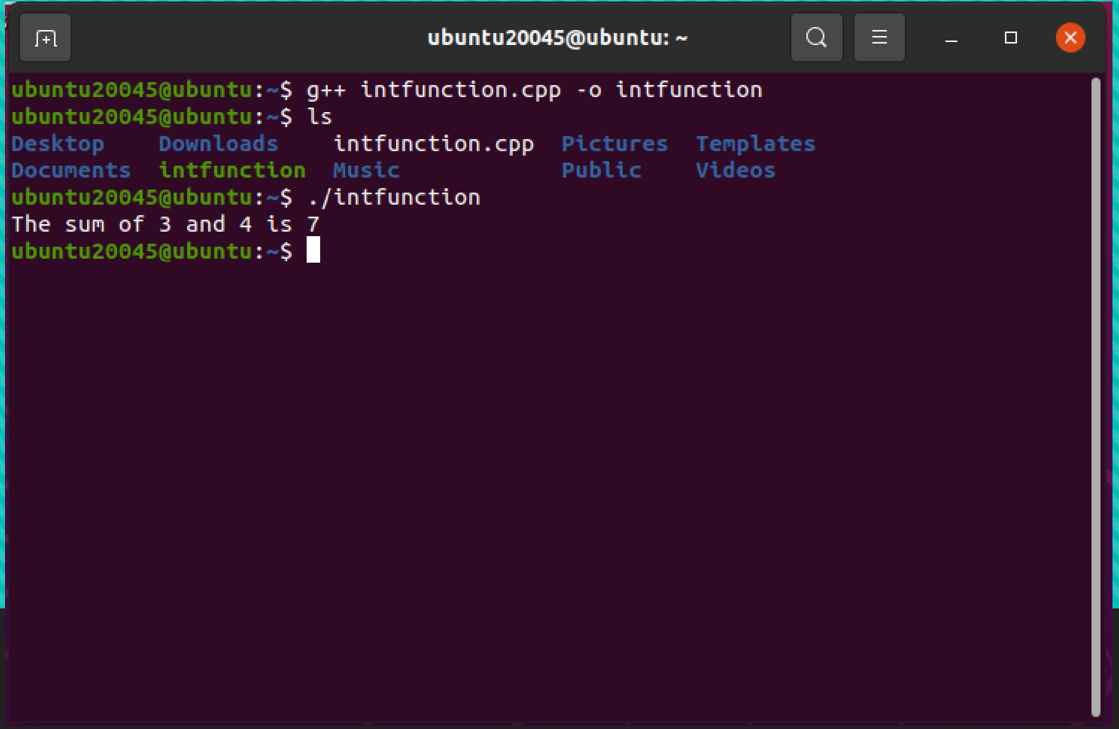
DASL-100.2 C++ Programming and Linux

Drones and Autonomous Systems Lab @ UNLV

1. Function



```cpp
#include <iostream>

int sum(int x, int y){
        return x + y;
};

int main(){
        int a = 3;
        int b = 4;
        int c = sum(a,b);
        std::cout << "The sum of " << a << " and " << b << " is " << c << std::endl;

        return 0;
};
```



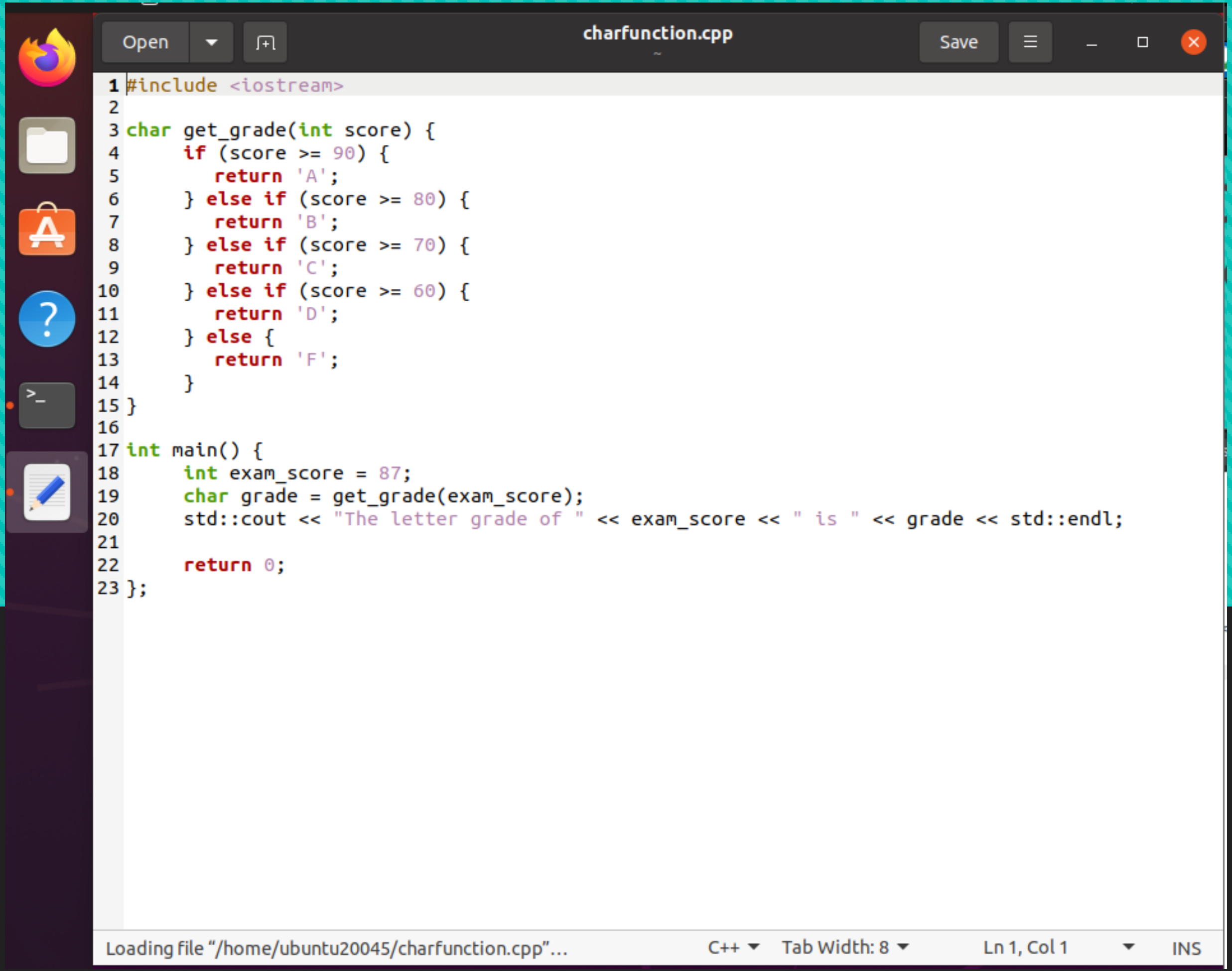
```
ubuntu20045@ubuntu:~$ g++ intfunction.cpp -o intfunction
ubuntu20045@ubuntu:~$ ls
Desktop    Downloads    intfunction.cpp  Pictures  Templates
Documents  intfunction  Music            Public    Videos
ubuntu20045@ubuntu:~$ ./intfunction
The sum of 3 and 4 is 7
ubuntu20045@ubuntu:~$ 
```

UNLV Howard R. Hughes College of Engineering

DASL-100.2 C++ Programming and Linux

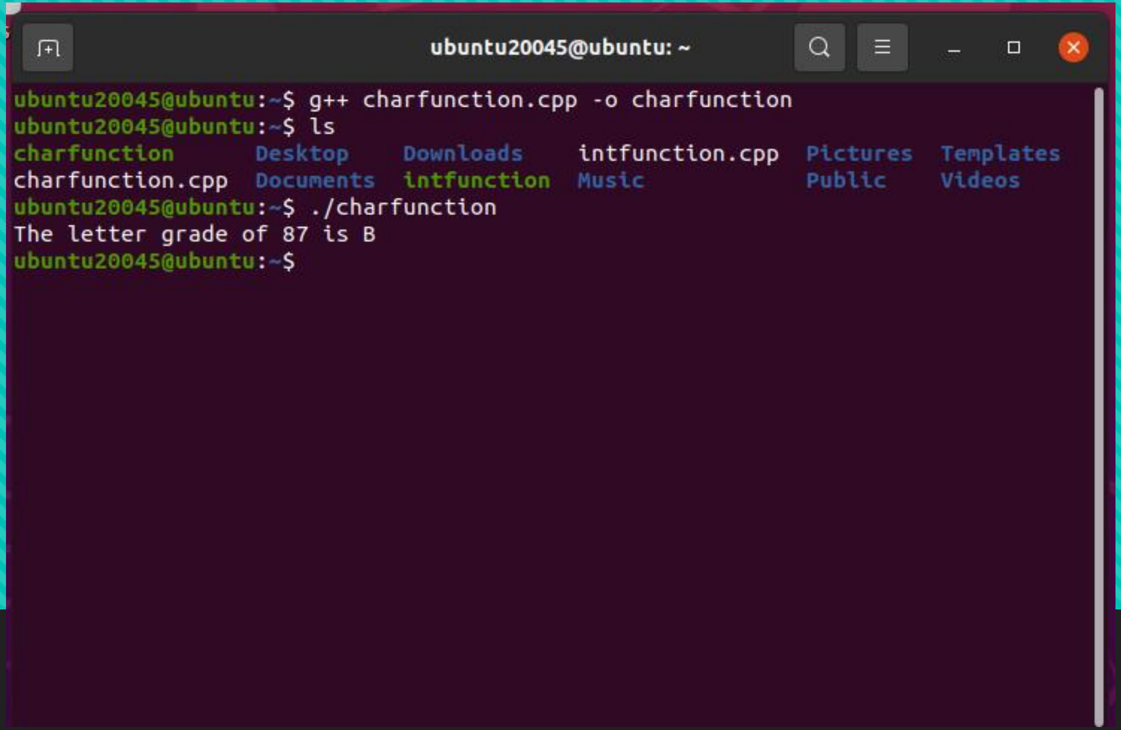
DASL Drones and Autonomous Systems Lab @ UNLV

1. Function



```cpp
#include <iostream>

char get_grade(int score) {
	if (score >= 90) {
	    return 'A';
	} else if (score >= 80) {
	    return 'B';
	} else if (score >= 70) {
	    return 'C';
	} else if (score >= 60) {
	    return 'D';
	} else {
	    return 'F';
	}
}

int main() {
	int exam_score = 87;
	char grade = get_grade(exam_score);
	std::cout << "The letter grade of " << exam_score << " is " << grade << std::endl;

	return 0;
};
```

Loading file "/home/ubuntu20045/charfunction.cpp"… C++ Tab Width: 8 Ln 1, Col 1 INS



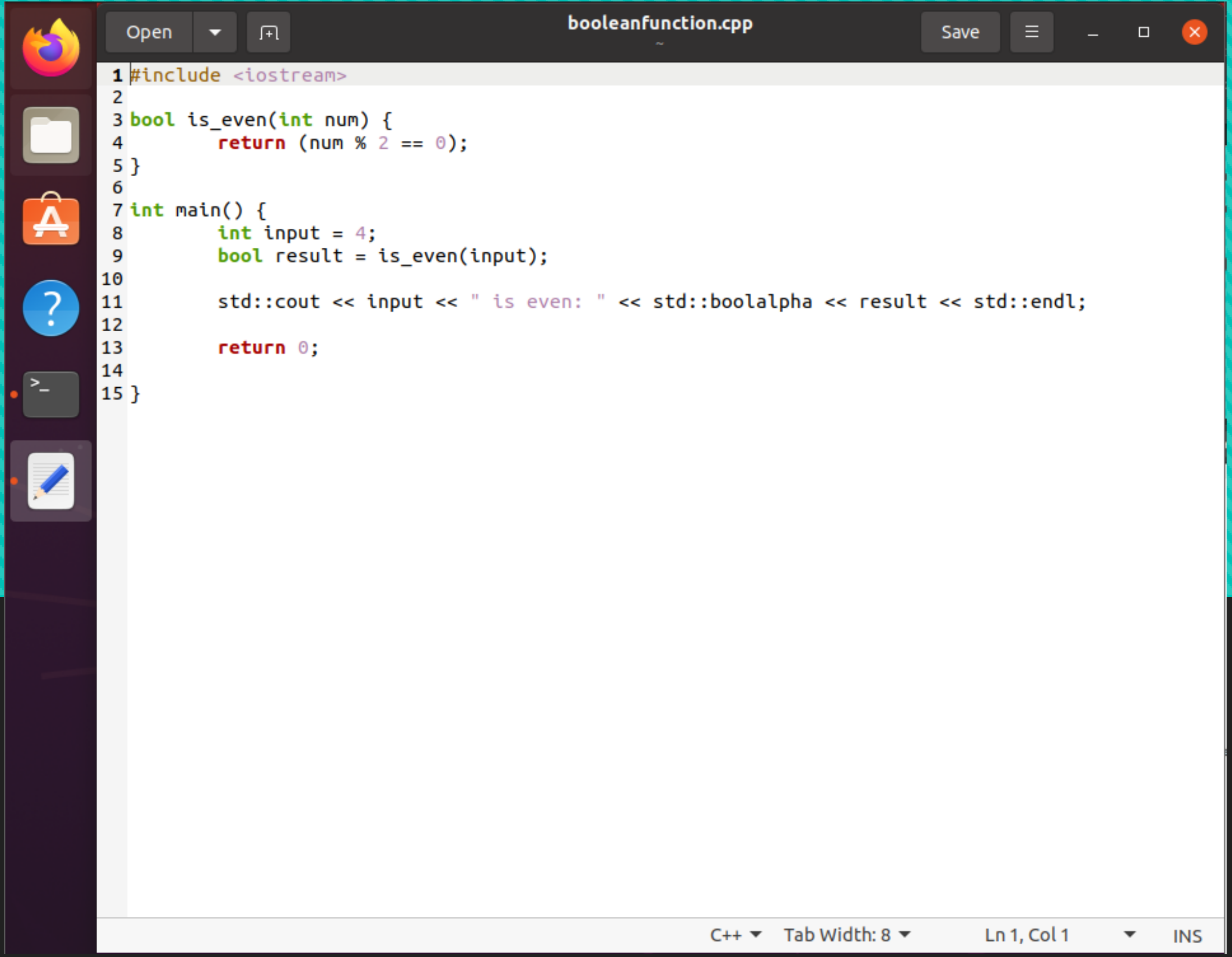
```
ubuntu20045@ubuntu:~$ g++ charfunction.cpp -o charfunction
ubuntu20045@ubuntu:~$ ls
charfunction      Desktop    Downloads    intfunction.cpp  Pictures  Templates
charfunction.cpp  Documents  intfunction  Music            Public    Videos
ubuntu20045@ubuntu:~$ ./charfunction
The letter grade of 87 is B
ubuntu20045@ubuntu:~$
```

UNLV | Howard R. Hughes College of Engineering

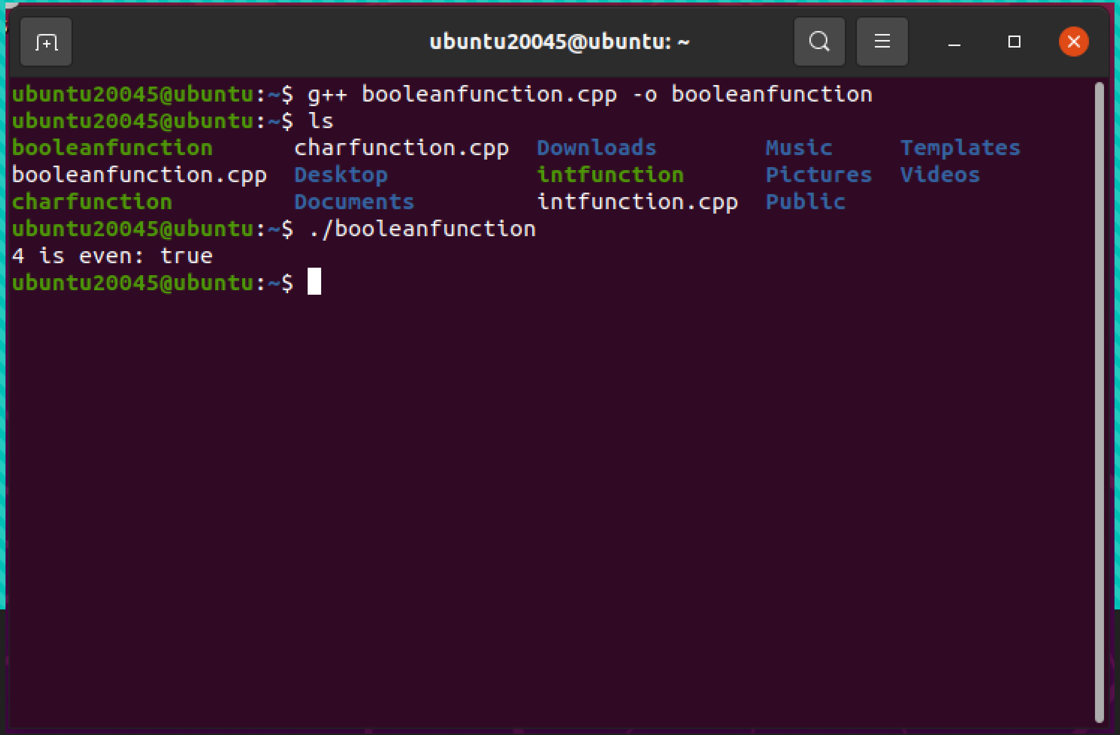
DASL-100.2 C++ Programming and Linux

DASL Drones and Autonomous Systems Lab @ UNLV

1. Function



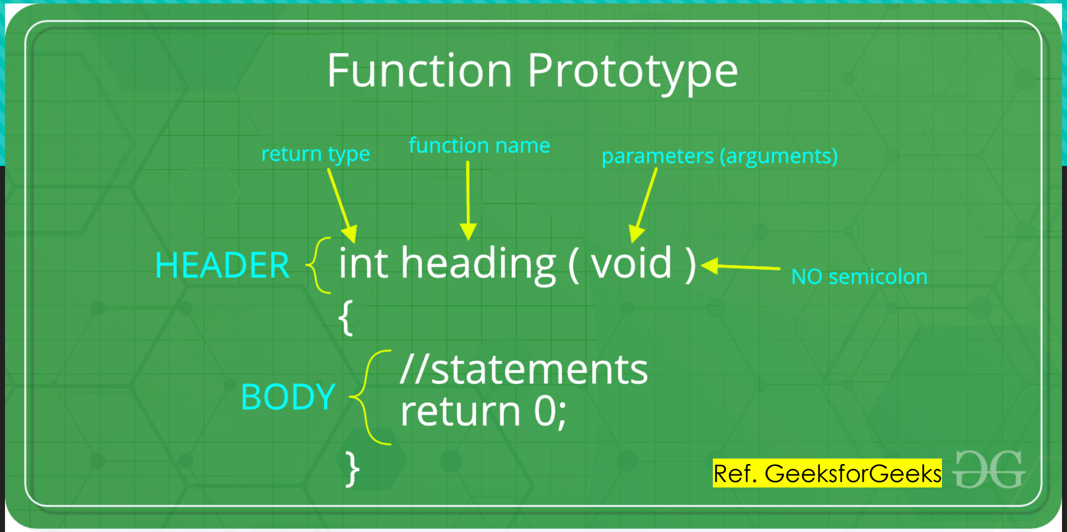
```cpp
#include <iostream>

bool is_even(int num) {
        return (num % 2 == 0);
}

int main() {
        int input = 4;
        bool result = is_even(input);

        std::cout << input << " is even: " << std::boolalpha << result << std::endl;

        return 0;

}
```



```
ubuntu20045@ubuntu:~$ g++ booleanfunction.cpp -o booleanfunction
ubuntu20045@ubuntu:~$ ls
booleanfunction      charfunction.cpp  Downloads        Music     Templates
booleanfunction.cpp  Desktop           intfunction      Pictures  Videos
charfunction         Documents         intfunction.cpp  Public
ubuntu20045@ubuntu:~$ ./booleanfunction
4 is even: true
ubuntu20045@ubuntu:~$ 
```

2. Structure

- In C++, a **struct** is a user-defined data type that groups together variables of different data types into a single unit. Each member of a **struct** has a name, and you can access its value using the dot operator (.)
- Unlike **array** is a way to store multiple values of the same data type in contiguous memory locations. Each element of an **array** has no name and just point to a value.
- In C++, structs do not have access control by default, meaning all members of a struct are public by default.



Function Prototype

```
int heading ( void )
{
    //statements
    return 0;
}
```

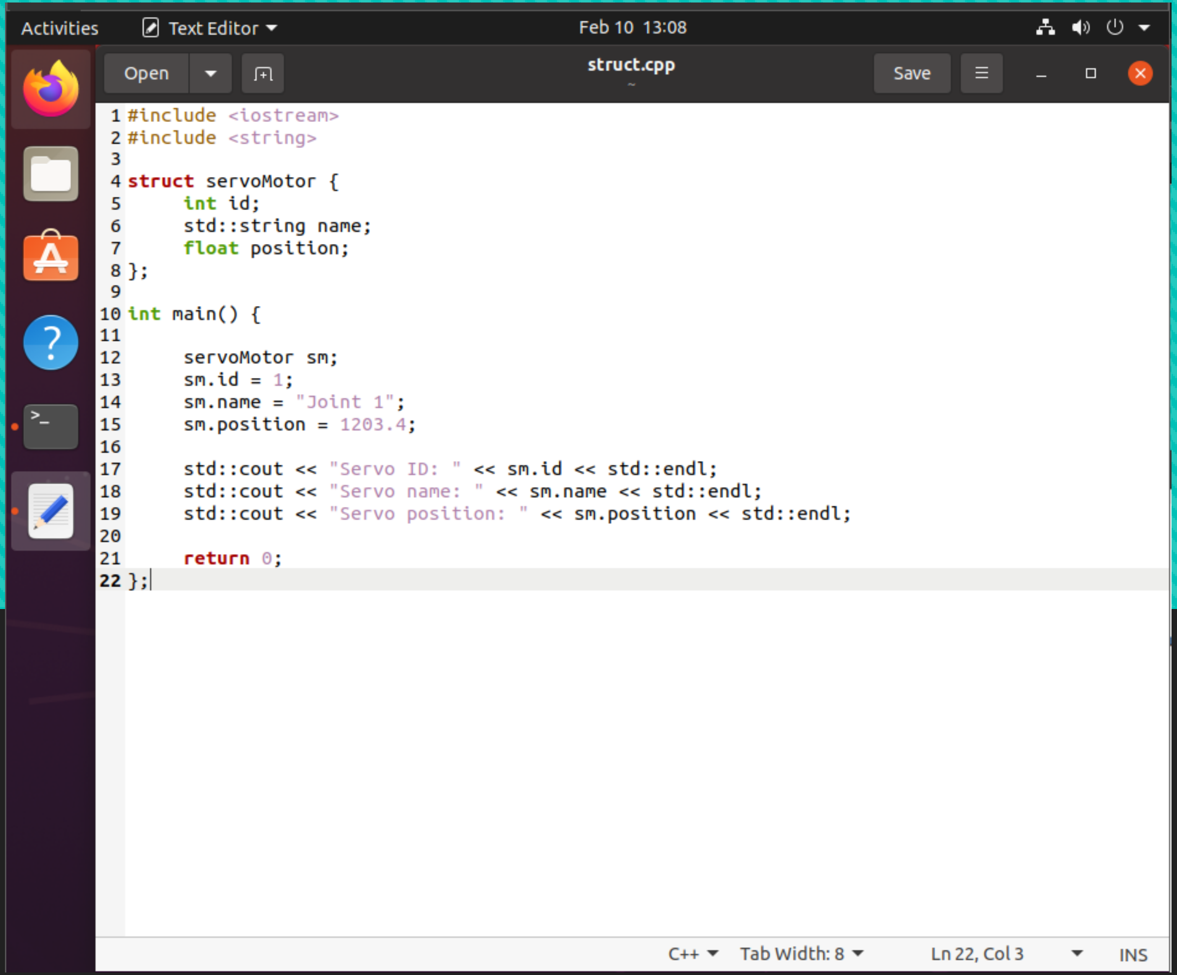
Ref. GeeksforGeeks

UNLV Howard R. Hughes College of Engineering

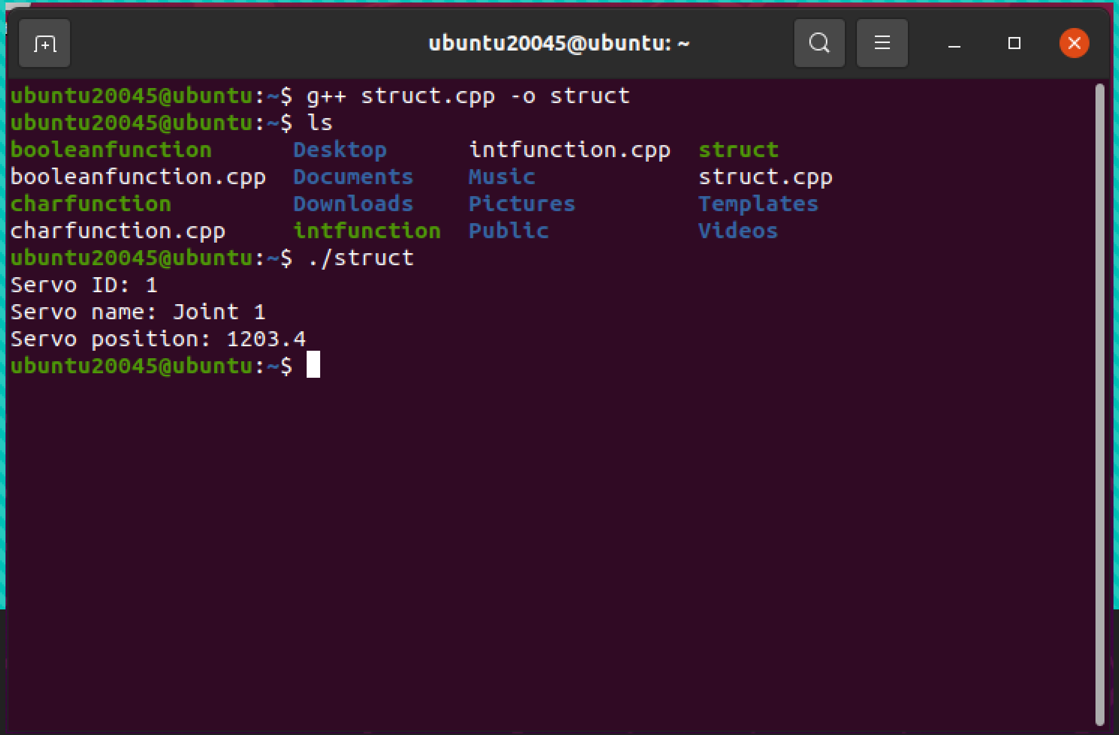
DASL-100.2 C++ Programming and Linux

DASL Drones and Autonomous Systems Lab @ UNLV

2. Structure



```cpp
#include <iostream>
#include <string>

struct servoMotor {
	int id;
	std::string name;
	float position;
};

int main() {

	servoMotor sm;
	sm.id = 1;
	sm.name = "Joint 1";
	sm.position = 1203.4;

	std::cout << "Servo ID: " << sm.id << std::endl;
	std::cout << "Servo name: " << sm.name << std::endl;
	std::cout << "Servo position: " << sm.position << std::endl;

	return 0;
};
```



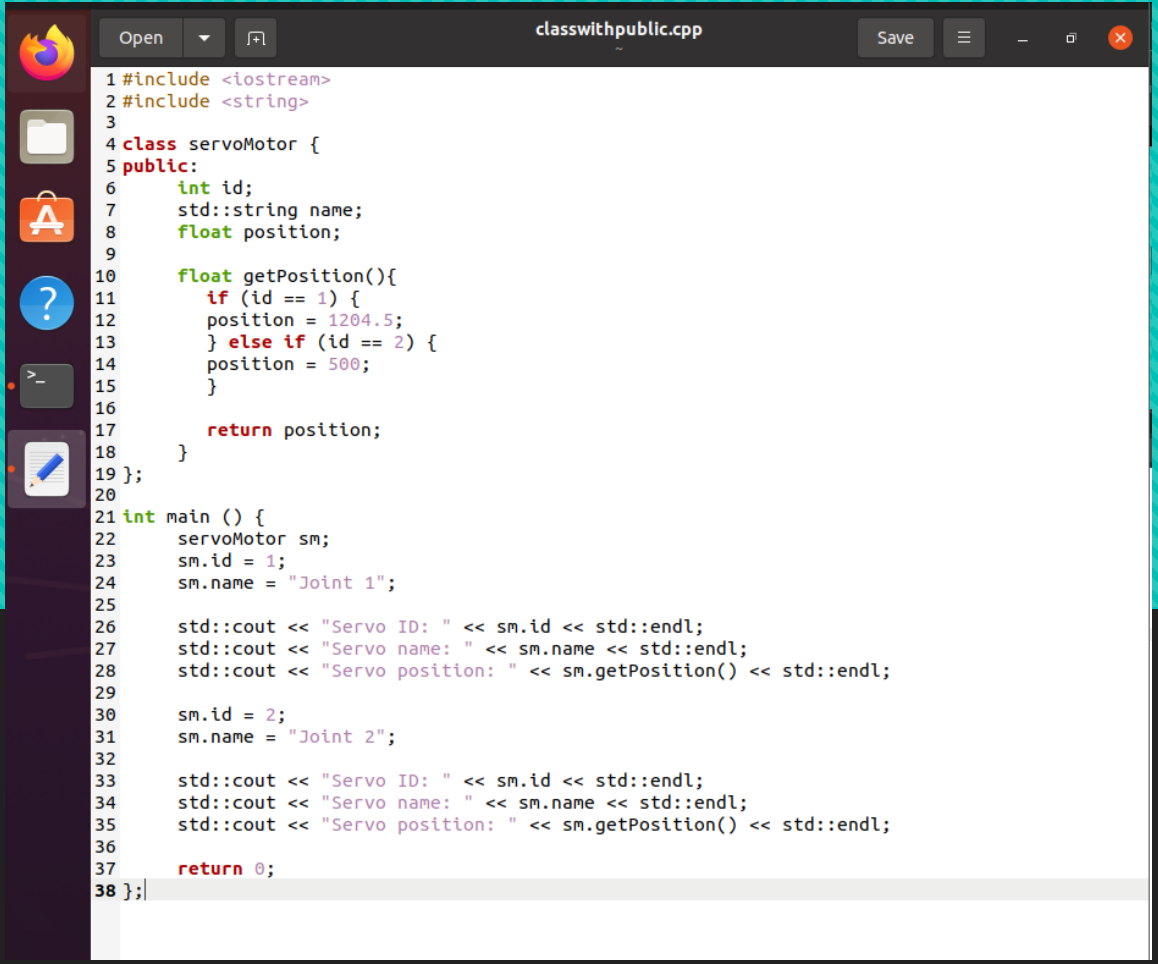
```
ubuntu20045@ubuntu:~$ g++ struct.cpp -o struct
ubuntu20045@ubuntu:~$ ls
booleanfunction       Desktop       intfunction.cpp  struct
booleanfunction.cpp   Documents     Music            struct.cpp
charfunction          Downloads     Pictures         Templates
charfunction.cpp      intfunction   Public           Videos
ubuntu20045@ubuntu:~$ ./struct
Servo ID: 1
Servo name: Joint 1
Servo position: 1203.4
ubuntu20045@ubuntu:~$
```

3. Class

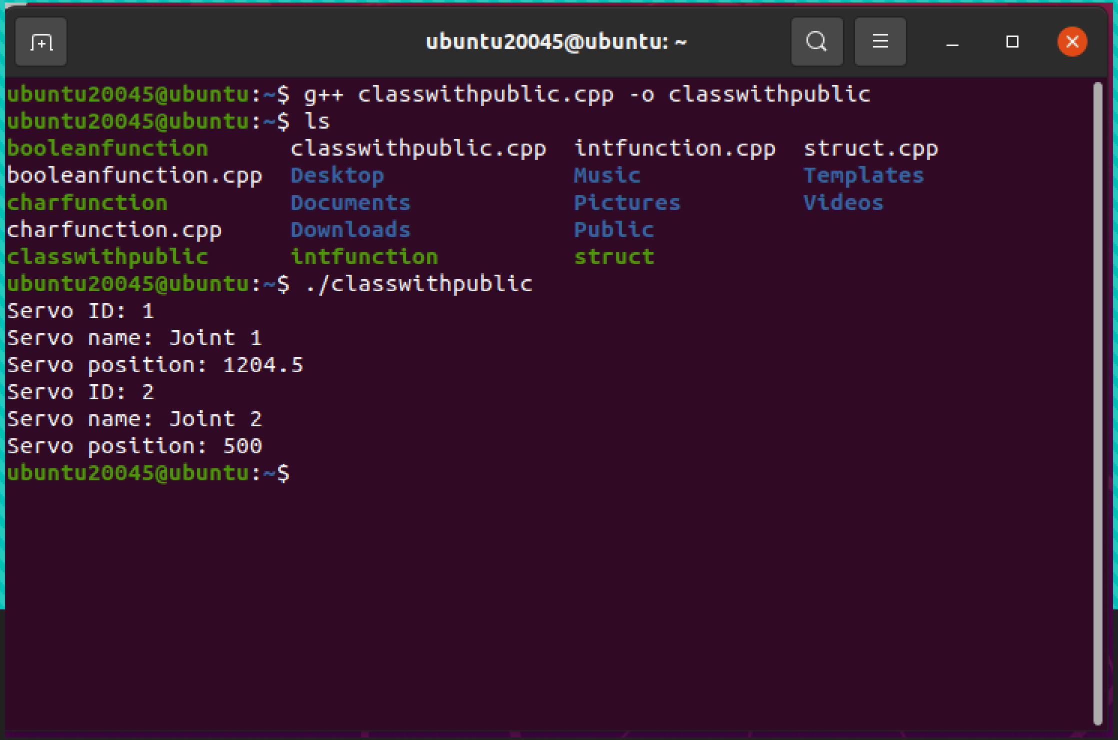
- In C++, a class is a user-defined data type that groups together variables (also known as data members) and functions (also known as member functions or methods) into a single unit
- A **class** is a blueprint for creating objects (instances of the class), and it encapsulates the data and behavior of the objects it creates.
- In C++, access control is an important aspect of object-oriented programming, and classes provide several access control mechanisms to restrict the visibility and accessibility of their members. These access control mechanisms are called access modifiers, and there are three types of access modifiers in C++:
  - **Public**: Members declared as public can be accessed from anywhere in the program. They are visible to the public and can be accessed from outside the class using the dot operator (.).
  - **Private**: Members declared as private can only be accessed from within the class. They are not visible to the public and cannot be accessed from outside the class.
  - **Protected**: Members declared as protected can only be accessed from within the class and its subclasses.

DASL-100.2
C++ Programming and Linux

3. Class



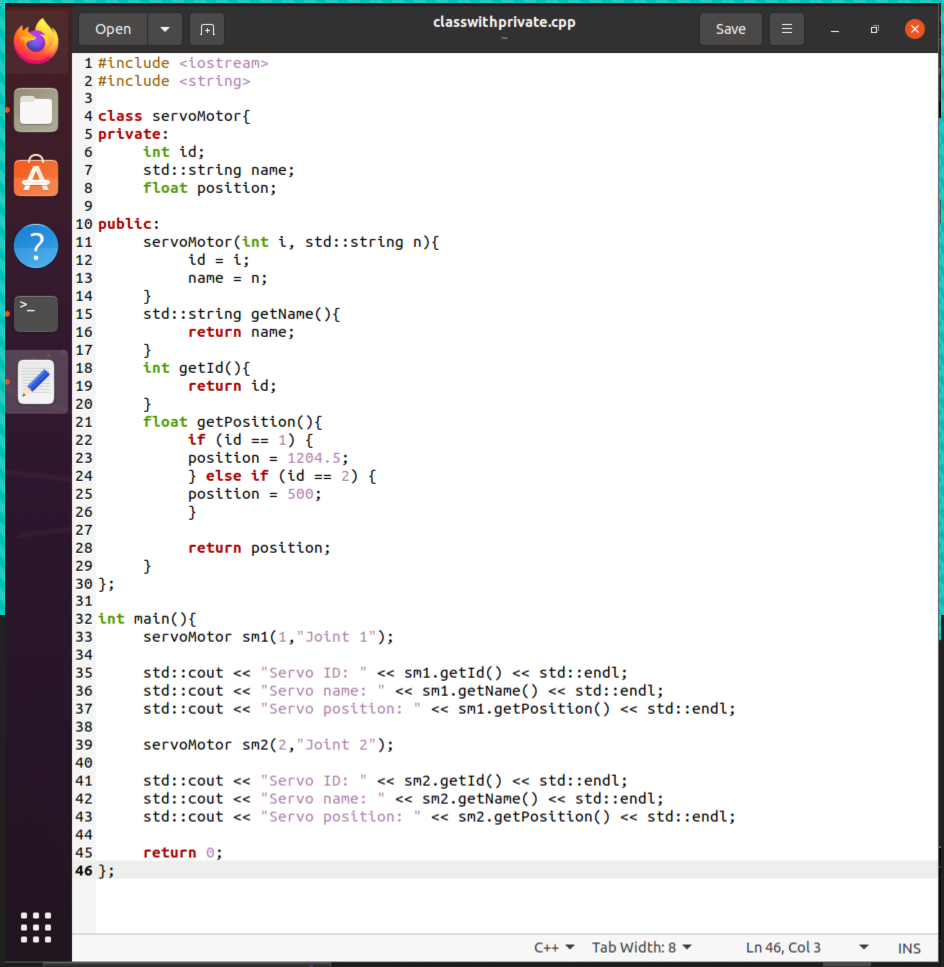
```cpp
#include <iostream>
#include <string>

class servoMotor {
public:
    int id;
    std::string name;
    float position;

    float getPosition(){
       if (id == 1) {
       position = 1204.5;
       } else if (id == 2) {
       position = 500;
       }

       return position;
    }
};

int main () {
    servoMotor sm;
    sm.id = 1;
    sm.name = "Joint 1";

    std::cout << "Servo ID: " << sm.id << std::endl;
    std::cout << "Servo name: " << sm.name << std::endl;
    std::cout << "Servo position: " << sm.getPosition() << std::endl;

    sm.id = 2;
    sm.name = "Joint 2";

    std::cout << "Servo ID: " << sm.id << std::endl;
    std::cout << "Servo name: " << sm.name << std::endl;
    std::cout << "Servo position: " << sm.getPosition() << std::endl;

    return 0;
};
```



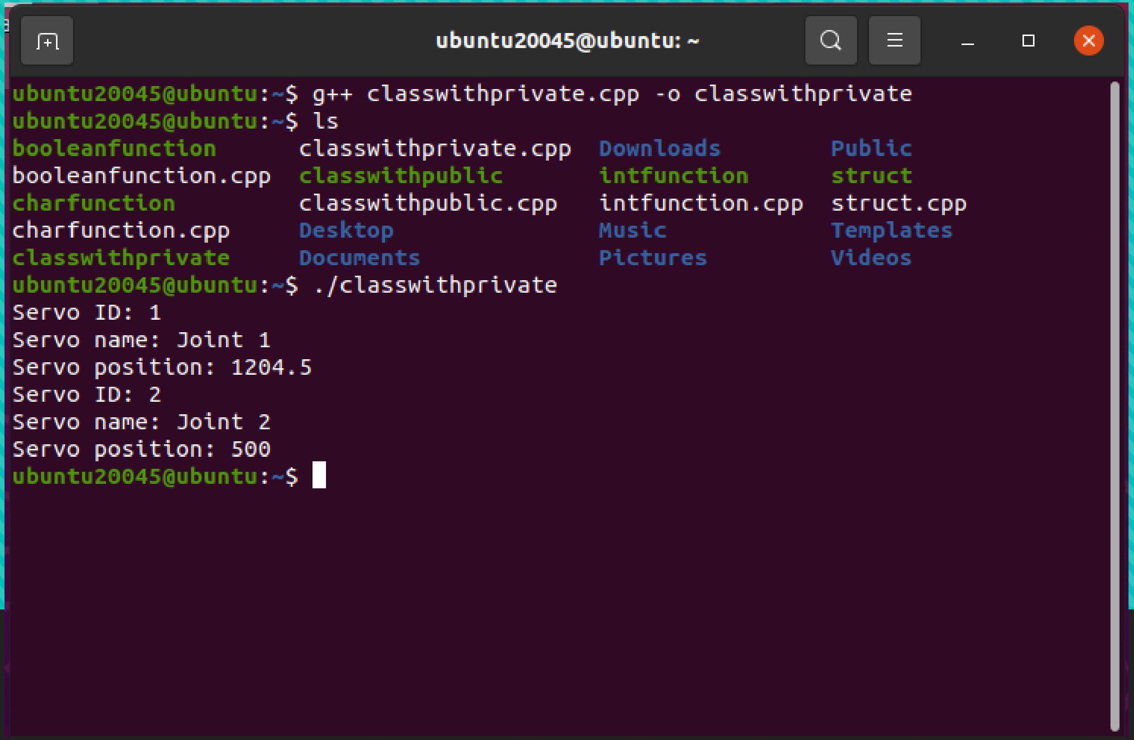
```
ubuntu20045@ubuntu:~$ g++ classwithpublic.cpp -o classwithpublic
ubuntu20045@ubuntu:~$ ls
booleanfunction      classwithpublic.cpp  intfunction.cpp  struct.cpp
booleanfunction.cpp  Desktop              Music            Templates
charfunction         Documents            Pictures         Videos
charfunction.cpp     Downloads            Public
classwithpublic      intfunction          struct
ubuntu20045@ubuntu:~$ ./classwithpublic
Servo ID: 1
Servo name: Joint 1
Servo position: 1204.5
Servo ID: 2
Servo name: Joint 2
Servo position: 500
ubuntu20045@ubuntu:~$
```

DASL-100.2
C++ Programming and Linux

3. Class



```cpp
#include <iostream>
#include <string>

class servoMotor{
private:
    int id;
    std::string name;
    float position;

public:
    servoMotor(int i, std::string n){
        id = i;
        name = n;
    }
    std::string getName(){
        return name;
    }
    int getId(){
        return id;
    }
    float getPosition(){
        if (id == 1) {
        position = 1204.5;
        } else if (id == 2) {
        position = 500;
        }

        return position;
    }
};

int main(){
    servoMotor sm1(1,"Joint 1");

    std::cout << "Servo ID: " << sm1.getId() << std::endl;
    std::cout << "Servo name: " << sm1.getName() << std::endl;
    std::cout << "Servo position: " << sm1.getPosition() << std::endl;

    servoMotor sm2(2,"Joint 2");

    std::cout << "Servo ID: " << sm2.getId() << std::endl;
    std::cout << "Servo name: " << sm2.getName() << std::endl;
    std::cout << "Servo position: " << sm2.getPosition() << std::endl;

    return 0;
};
```



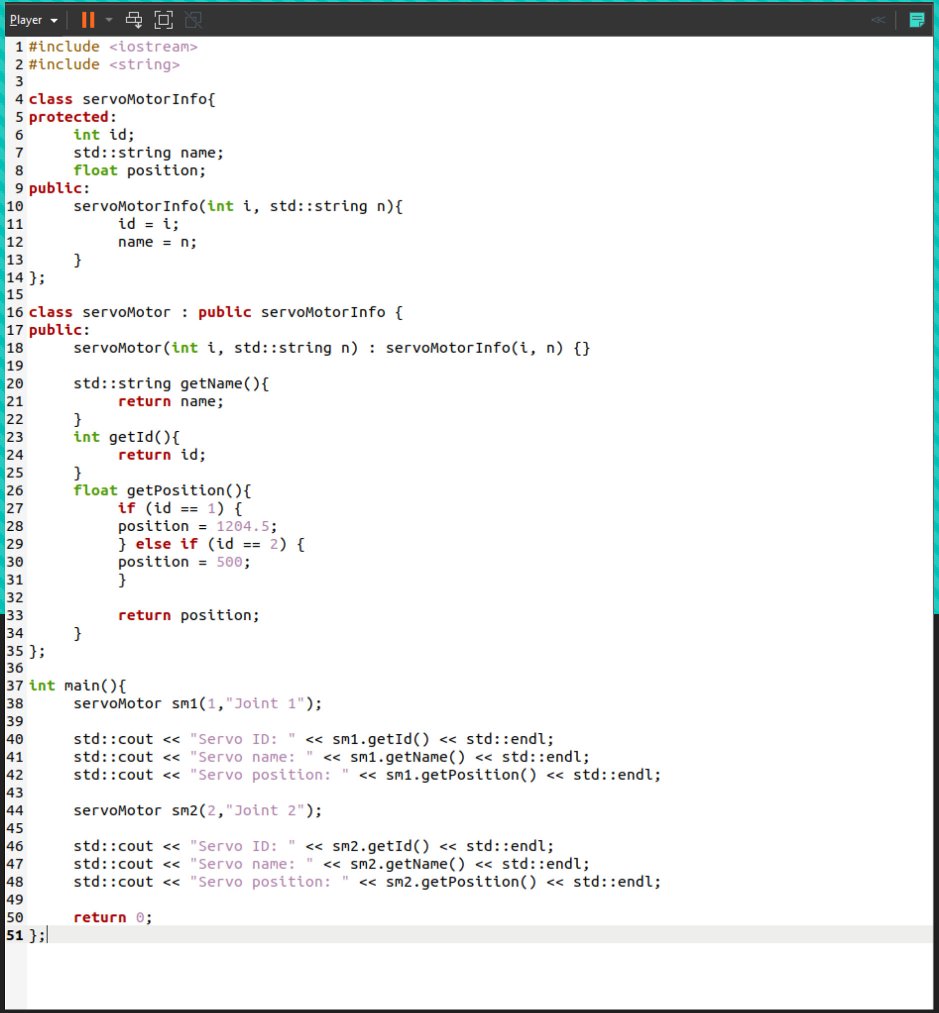
```
ubuntu20045@ubuntu:~$ g++ classwithprivate.cpp -o classwithprivate
ubuntu20045@ubuntu:~$ ls
booleanfunction      classwithprivate.cpp  Downloads        Public
booleanfunction.cpp  classwithpublic       intfunction      struct
charfunction         classwithpublic.cpp   intfunction.cpp  struct.cpp
charfunction.cpp     Desktop               Music            Templates
classwithprivate     Documents             Pictures         Videos
ubuntu20045@ubuntu:~$ ./classwithprivate
Servo ID: 1
Servo name: Joint 1
Servo position: 1204.5
Servo ID: 2
Servo name: Joint 2
Servo position: 500
ubuntu20045@ubuntu:~$ 
```

UNLV Howard R. Hughes College of Engineering

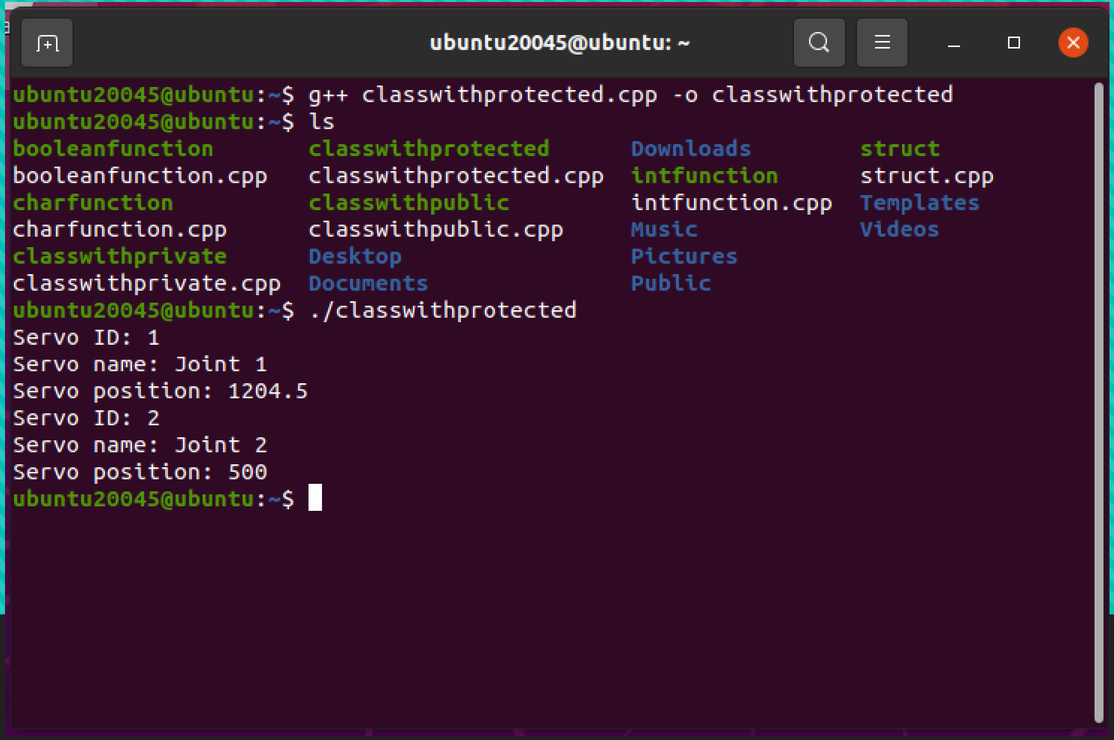
DASL-100.2 C++ Programming and Linux

DASL Drones and Autonomous Systems Lab @ UNLV

3. Class



```cpp
#include <iostream>
#include <string>

class servoMotorInfo{
protected:
    int id;
    std::string name;
    float position;
public:
    servoMotorInfo(int i, std::string n){
        id = i;
        name = n;
    }
};

class servoMotor : public servoMotorInfo {
public:
    servoMotor(int i, std::string n) : servoMotorInfo(i, n) {}

    std::string getName(){
        return name;
    }
    int getId(){
        return id;
    }
    float getPosition(){
        if (id == 1) {
        position = 1204.5;
        } else if (id == 2) {
        position = 500;
        }

        return position;
    }
};

int main(){
    servoMotor sm1(1,"Joint 1");

    std::cout << "Servo ID: " << sm1.getId() << std::endl;
    std::cout << "Servo name: " << sm1.getName() << std::endl;
    std::cout << "Servo position: " << sm1.getPosition() << std::endl;

    servoMotor sm2(2,"Joint 2");

    std::cout << "Servo ID: " << sm2.getId() << std::endl;
    std::cout << "Servo name: " << sm2.getName() << std::endl;
    std::cout << "Servo position: " << sm2.getPosition() << std::endl;

    return 0;
};
```



```
ubuntu20045@ubuntu:~$ g++ classwithprotected.cpp -o classwithprotected
ubuntu20045@ubuntu:~$ ls
booleanfunction       classwithprotected      Downloads         struct
booleanfunction.cpp   classwithprotected.cpp  intfunction       struct.cpp
charfunction          classwithpublic         intfunction.cpp   Templates
charfunction.cpp      classwithpublic.cpp     Music             Videos
classwithprivate      Desktop                 Pictures
classwithprivate.cpp  Documents               Public
ubuntu20045@ubuntu:~$ ./classwithprotected
Servo ID: 1
Servo name: Joint 1
Servo position: 1204.5
Servo ID: 2
Servo name: Joint 2
Servo position: 500
ubuntu20045@ubuntu:~$
```