

## - Solution

This lesson explains the solution for the exercise in the previous lesson.

### WE'LL COVER THE FOLLOWING ^

- Solution
- Explanation

## Solution #

```
struct OnlyInt{
    OnlyInt(int){}

    template<typename T>
    OnlyInt(T) = delete;
};

int main(){

    OnlyInt(5);
    OnlyInt(5L);
    OnlyInt(5LL);
    OnlyInt(5UL);
    OnlyInt(5.5);
    OnlyInt('5');
    OnlyInt(true);

}
```



## Explanation #

- This struct is very similar to the `onlyDouble` struct we made in the last [lesson](#). The only difference is that it accepts `int`.
- We define a constructor with an `int` parameter, as demonstrated in line 2.

- In the template, we assign `delete` to the constructor with any type other than `int`. This tells the compiler to reject any other data types as arguments.
  - Lines 12 to 17 demonstrate several data types that would cause the compiler to throw an error.
- 

That brings us to the end of our discussion on methods. The next concept we will tackle is **operator overloading**.