

Basic C++

```
// CSCI 159  
// Sept 9, 2024  
// Gara Pruesse
```

Notes

```
// makes everything to the  
right on the same line  
into a comment
```

```
int main() {  
    int x;  
    int y;  
    x = 5;  
    y = 7;  
    y = y + 5;  
    return 0;  
}
```

Basic C++

This is the "return type" - it is "the kind of data item that main returns"

↓
`int main() {`
 `int x;`
 `int y;`
 `x = 5;`
 `y = 7;`
 `y = y + 5;`
 `return 0;`
`}`

← this is what is returned - it has to match the returned type.

Our compiler doesn't mind if "main" does not return an integer value - but other compilers may. So do it!

C++ Types

```
int main() {
```

```
    int x;
```

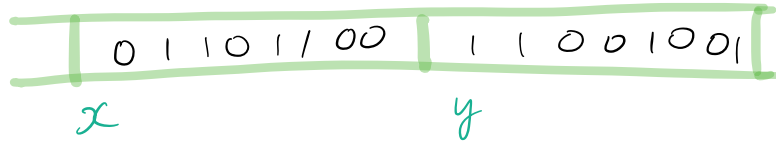
```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + 5;
```

```
}
```



C++ Types

```
int main() {
```

```
    int x;
```

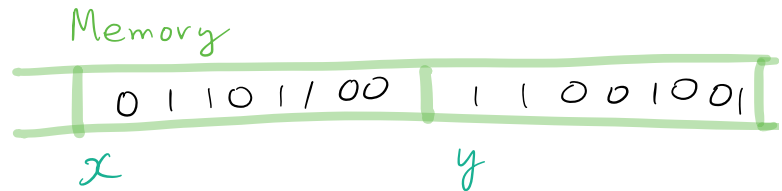
```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + 5;
```

```
}
```



This is called "declaring variables x and y "

- it means that space in **Memory** is set aside to store the values of things (data objects) that we will refer to as x and y .

C++ Types

```
int main() {
```

```
int x;
```

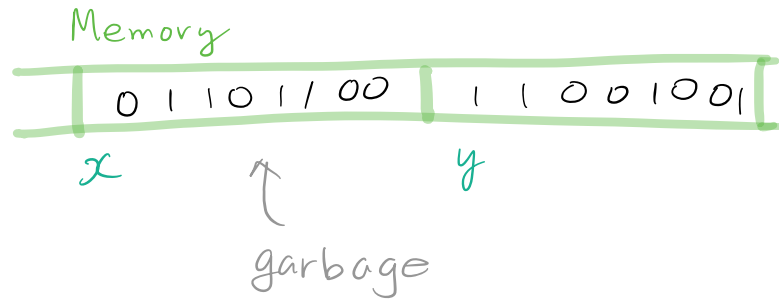
```
int y;
```

```
x = 5;
```

```
y = 7;
```

```
y = y + 5;
```

```
}
```



C++ Types and Assignment Statements

```
int main() {
```

```
    int x;
```

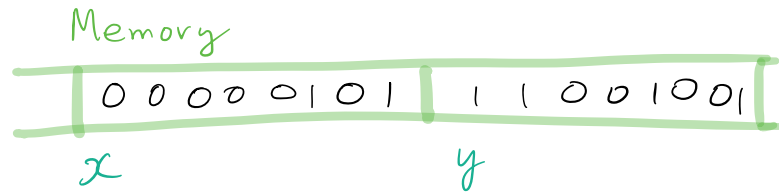
```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + 5;
```

```
}
```



C++ Types and Assignment Statements

```
int main() {
```

```
    int x;
```

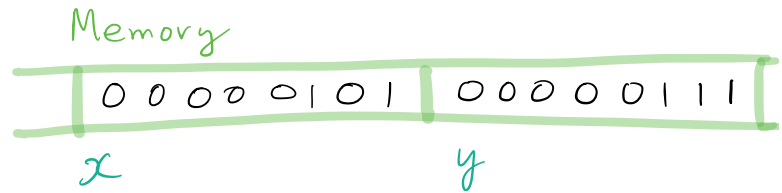
```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + 5;
```

```
}
```



C++ Types and Assignment Statements

```
int main() {
```

```
    int x;
```

```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

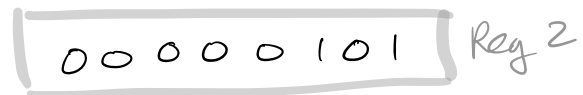
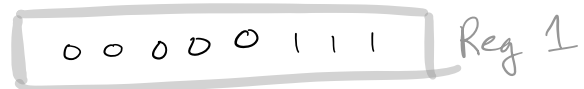
```
    y = y + 5;
```

```
}
```

Memory



Arithmetic is done in the Arithmetic Logic Unit, not in memory



C++ Types and Assignment Statements

```
int main() {
```

```
int x;
```

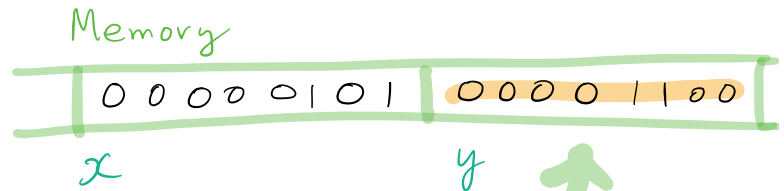
```
int y;
```

```
x = 5;
```

```
y = 7;
```

```
y = y + 5;
```

```
}
```



Arithmetic is done in the Arithmetic Logic Unit, not in memory

00000111 Reg 1

00000101 Reg 2

00001100 Result Register

C++ Types and Assignment Statements

```
int main() {
```

```
int x;
```

```
int y;
```

```
x = 5;
```

```
y = 7;
```

```
y = y + 5;
```

```
}
```

Memory



// This is not an assertion

this is an imperative

("add 5 to the value
in variable y,
and store the result
back in y")

C++ Types and Assignment Statements

```
int main() {
```

```
    int x;
```

```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + x;
```

```
}
```

Memory



// variables can also appear
on the RHS of
assignment statements

C++ Types and Assignment Statements

```
int main() {
```

```
    int x;
```

```
    int y;
```

```
    x = 5;
```

```
    y = 7;
```

```
    y = y + x;
```

```
}
```

Memory



Why does a variable have to have a type?

- so the compiler sets aside the right amount of memory for this object

- so the compiler can