Data

- int
- float
- double
- char
- string

Expressions

- I/O expressions
- Arithmetic expressions

Control Flow

- Sequential

Data

- int
- float
- double
- char
- string
- bool

Expressions

- I/O expressions
- Arithmetic expressions

Control Flow

- Sequential

The `bool` Data Type

The `bool` Data Type

Kind of data:

The `bool` Data Type

Kind of data: Truth value (True/False)

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals:

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators:

Not

Not

p	**not** p
True	False
False	True

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !

Not

p	**not** p
True	False
False	True

```
int main(){
        bool b1, b2, b3;




        return 0;
}
```

Not

p	**not** p
True	False
False	True

```
int main(){
        bool b1, b2, b3;

        b1 = true;



        return 0;
}
```

Not

p	**not** p
True	False
False	True

```
int main(){
        bool b1, b2, b3;

        b1 = true;
        b2 = !b1;


        return 0;
}
```

Not

p	**not** p
True	False
False	True

```
int main(){
        bool b1, b2, b3;

        b1 = true;
        b2 = !b1;
        b3 = !false;

        return 0;
}
```

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !

And

And

p	q	p **and** q
True	True	True
True	False	False
False	True	False
False	False	False

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !, &&

And

p	q	p **and** q
True	True	True
True	False	False
False	True	False
False	False	False

```
int main(){
        bool b1, b2, b3;









        return 0;
}
```

And

p	q	p **and** q
True	True	True
True	False	False
False	True	False
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = true;
        b2 = false;



        return 0;
}
```

And

p	q	p **and** q
True	True	True
True	False	False
False	True	False
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = true;
        b2 = false;
        b3 = b1 && b2;


        return 0;
}
```

And

p	q	p **and** q
True	True	True
True	False	False
False	True	False
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = true;
        b2 = false;
        b3 = b1 && b2;
        b3 = b1 && !b2;

        return 0;
}
```

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !, &&

Or

Or

p	q	p **or** q
True	True	True
True	False	True
False	True	True
False	False	False

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !, &&, ||

Or

p	q	p **or** q
True	True	True
True	False	True
False	True	True
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = false;



        return 0;
}
```

Or

p	q	p **or** q
True	True	True
True	False	True
False	True	True
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = false;
        b2 = b1 || !b1;


        return 0;
}
```

Or

p	q	p **or** q
True	True	True
True	False	True
False	True	True
False	False	False

```
int main(){
        bool b1, b2, b3;

        b1 = false;
        b2 = b1 || !b1;
        b3 = b2 && (b1 || true);

        return 0;
}
```

The `bool` Data Type

Kind of data: Truth value (True/False)

Inner representation:

- Each `bool` data uses 1 byte (8 bits)
- False is represented by a byte of 0s
  True is any non-zero value

C++ literals: true, false

Boolean Operators: !, &&, ||

Data

- int
- float
- double
- char
- string
- bool

Expressions

- I/O expressions
- Arithmetic expressions

Control Flow

- Sequential

Data

- int
- float
- double
- char
- string
- bool

Expressions

- I/O expressions
- Arithmetic expressions
- Boolean expressions

Control Flow

- Sequential

Boolean Expressions

Boolean Expressions

Atomic Boolean Expressions:

Compound Boolean Expressions:

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**

Compound Boolean Expressions:

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

```
int main(){
        bool b;









        return 0;
}
```

```
int main(){
	bool b;


	b = true;









	return 0;
}
```

```
int main(){
	bool b;


	b = true;
	b = (true && !b);









	return 0;
}
```

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**
- Arithmetic expressions compared with relational operators (<, >, <=, >=,)

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

```
int main(){
	bool b;


	b = true;
	b = (true && !b);









	return 0;
}
```

```
int main(){
	bool b;
	int x;

	b = true;
	b = (true && !b);

	x = 3;



	return 0;
}
```

```
int main(){
	bool b;
	int x;

	b = true;
	b = (true && !b);

	x = 3;
	b = (x < 5);



	return 0;
}
```

```
int main(){
	bool b;
	int x;

	b = true;
	b = (true && !b);

	x = 3;
	b = (x < 5);
	b = (x >= 0) && (x < 5);


	return 0;
}
```

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**
- Arithmetic expressions compared with relational operators (<, >, <=, >=)

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**
- Arithmetic expressions compared with relational operators (<, >, <=, >=, ==)

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

Boolean Expressions

Atomic Boolean Expressions:

- The `bool` literals – **`true`**, **`false`**
- Arithmetic expressions compared with relational operators (<, >, <=, >=, ==, !=)

Compound Boolean Expressions:

Simple boolean expressions combined with boolean operators (!, &&, ||)

```
int main(){
	bool b;
	int x;

	b = true;
	b = (true && !b);

	x = 3;
	b = (x < 5);
	b = (x >= 0) && (x < 5);


	return 0;
}
```

```
int main(){
	bool b;
	int x;

	b = true;
	b = (true && !b);

	x = 3;
	b = (x < 5);
	b = (x >= 0) && (x < 5);
	b = (x == 3) || (x == 4);

	return 0;
}
```