

Career Explorations

Programming 101

Eric and Anshuman, June 28 2022

Variables, values, and types

```
int a = 32
```

Variables, values, and types

```
int a = 32
```

variable

Variables, values, and types

```
int  a  =  32
```

type variable

Variables, values, and types

```
int  a  =  32
```

type variable value

Variables, values, and types

```
int  a  =  32
```

type variable value

```
int  b  =  12
```

Variables, values, and types

```
int  a  =  32
  type variable value
```

```
int  b  =  12
```

a

Variables, values, and types

```
int  a  =  32
  type variable value
```

```
int  b  =  12
```

a

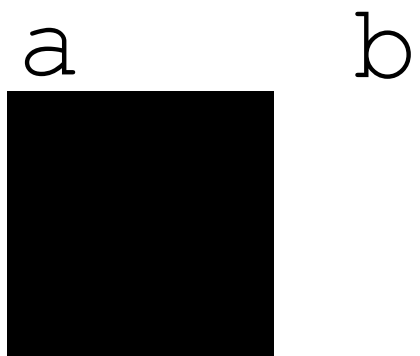
b

Variables, values, and types

```
int  a  = 32
```

type variable value

```
int  b  = 12
```

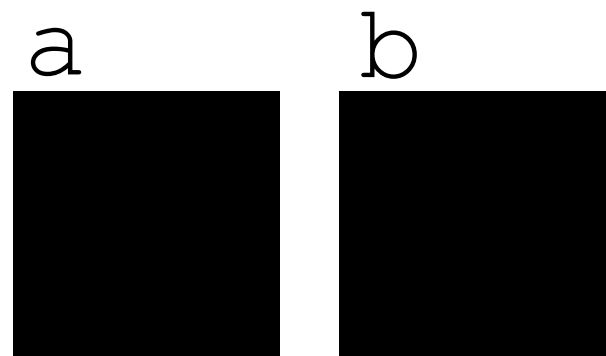


Variables, values, and types

```
int  a  = 32
```

type variable value

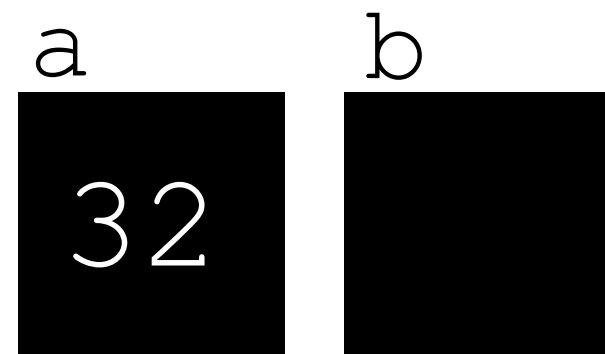
```
int  b  = 12
```



Variables, values, and types

```
int  a  =  32
  type variable value
```

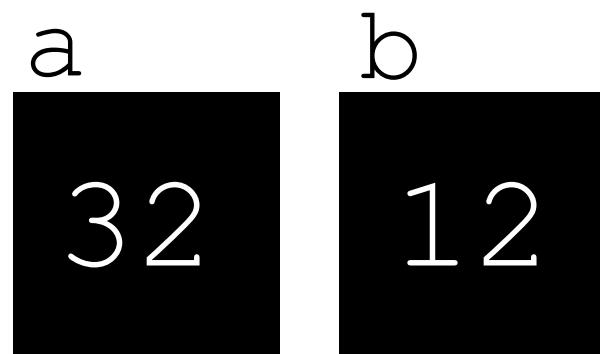
```
int  b  =  12
```



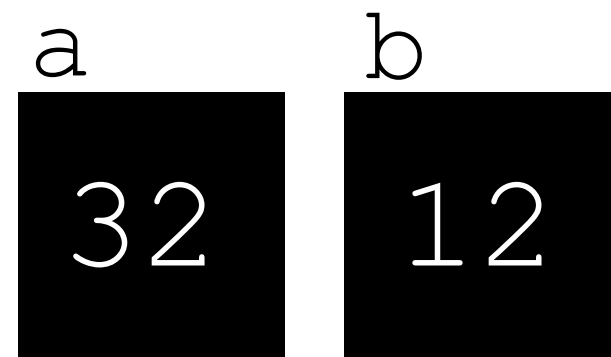
Variables, values, and types

```
int  a  =  32
  type variable value
```

```
int  b  =  12
```



Variables, values, and types

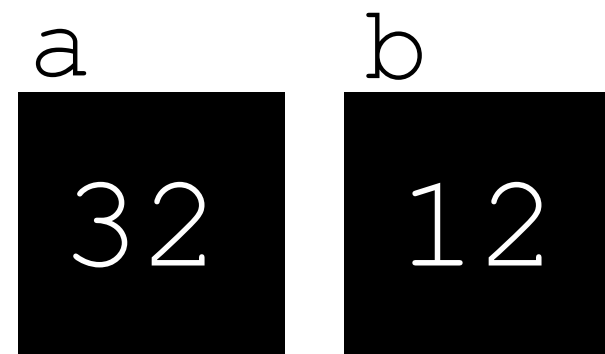


```
int  a  =  32
  type variable value
```

```
int  b  =  12
```

```
int  c  =  a-b
```

Variables, values, and types

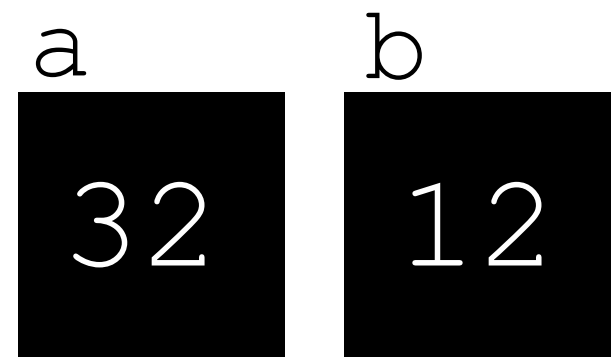


`int a = 32`
type variable value

`int b = 12`

`int c = a - b`

Variables, values, and types



```
int a = 32
```

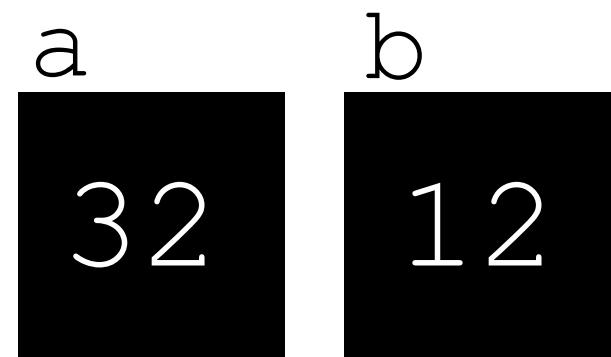
type variable value

```
int b = 12
```

```
int c = a - b
```

32

Variables, values, and types



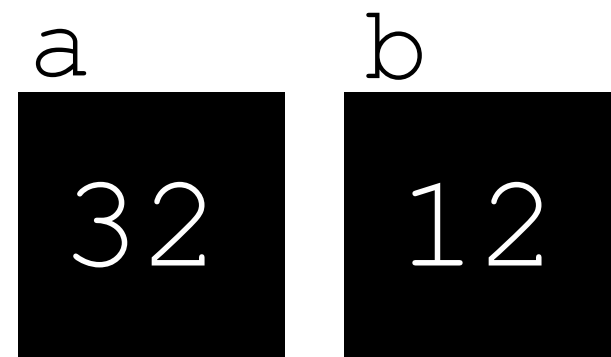
`int a = 32`
type variable value

`int b = 12`

`int c = a - b`
32

A diagram showing the expression 'a - b' from the code line 'int c = a - b'. Two red arrows originate from below the expression: one points to the 'a' and the other points to the 'b'. Above the 'a' is the number '32' in red, indicating its value.

Variables, values, and types



```
int a = 32
```

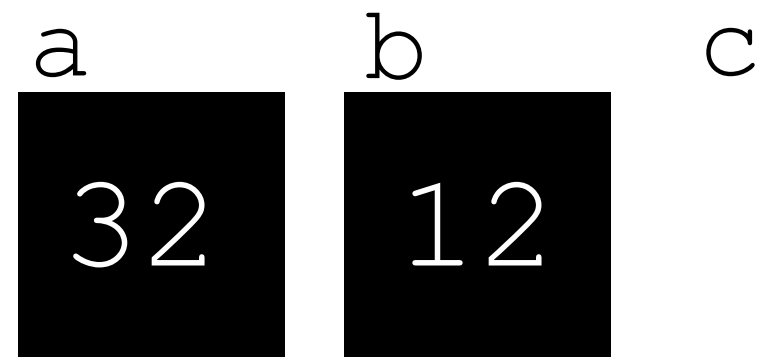
type variable value

```
int b = 12
```

```
int c = a - b
```

Diagram illustrating the expression `a - b` in the assignment `int c = a - b`. Red arrows point from the variable `a` to the value 32 and from the variable `b` to the value 12, indicating the values used in the calculation.

Variables, values, and types



```
int  a  =  32
```

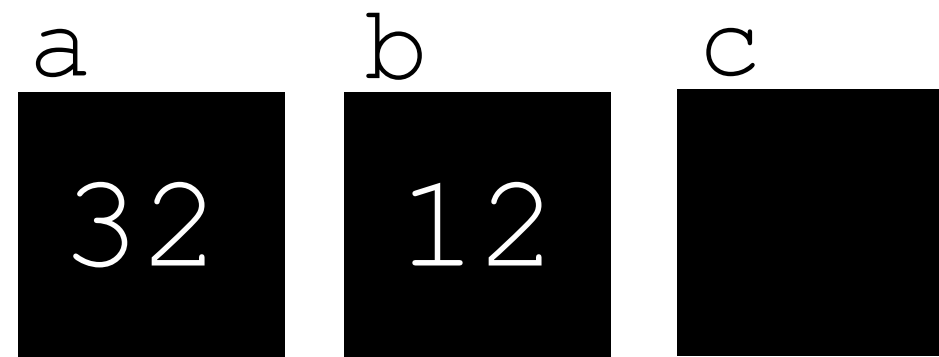
type variable value

```
int  b  =  12
```

```
int  c  =  a - b
```

Diagram illustrating the assignment statement `int c = a - b`. Red arrows point from the variable `a` to the value 32 and from the variable `b` to the value 12, indicating the values used in the calculation.

Variables, values, and types

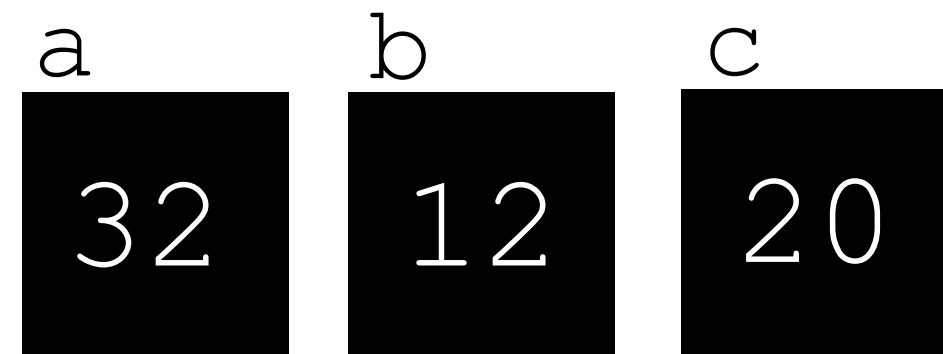


`int a = 32`
type variable value

`int b = 12`

`int c = a - b`
32 12

Variables, values, and types

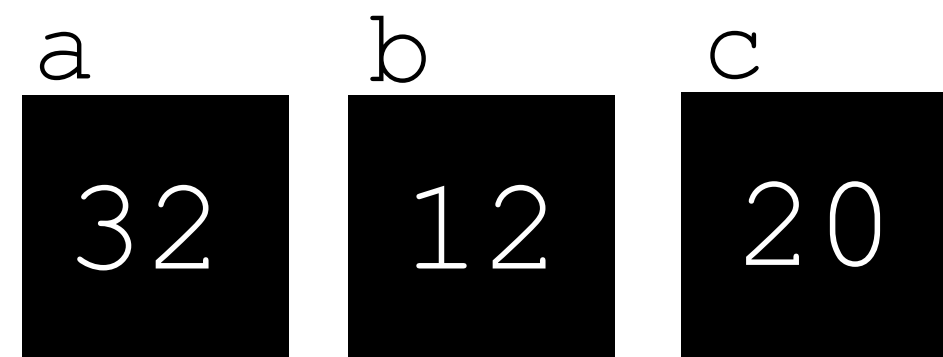


`int a = 32`
type variable value

`int b = 12`

`int c = a - b`
Diagram illustrating the calculation of the value for variable c. Red arrows point from the variable 'a' in the expression 'a - b' to the value '32' (above 'a') and from the variable 'b' to the value '12' (above 'b').

Variables, values, and types



```
int a = 32
```

type variable value

```
int b = 12
```

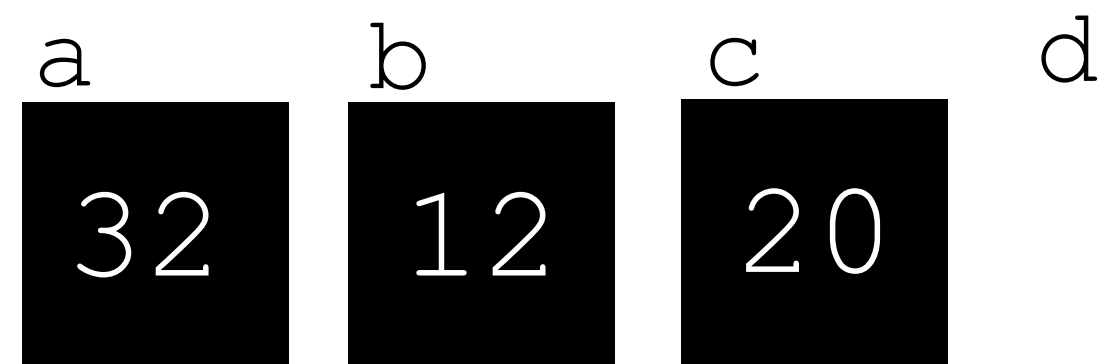
```
int c = a - b
```

Diagram illustrating the calculation of `a - b`:

- Red arrow from `a` to `32`
- Red arrow from `b` to `12`

```
bool d = a > b
```

Variables, values, and types



```
int a = 32
```

type variable value

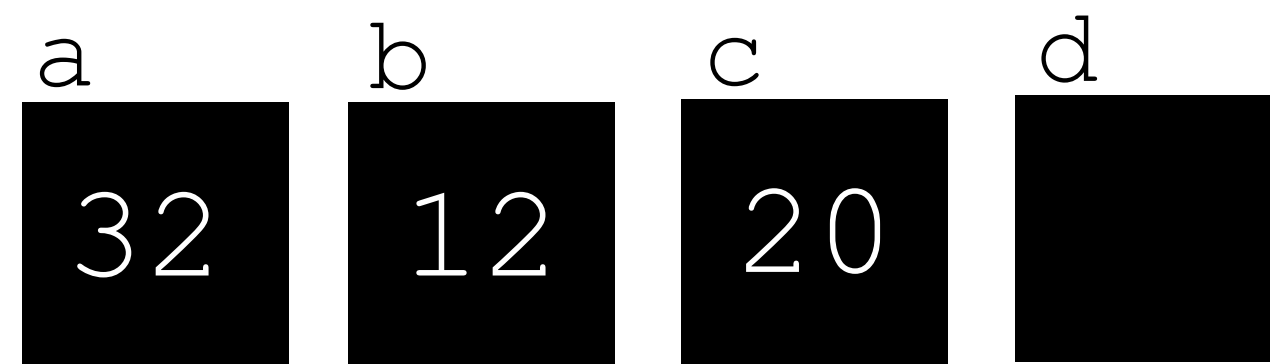
```
int b = 12
```

```
int c = a - b
```

Diagram illustrating the calculation of the value for variable c. Red arrows point from the values 32 (from variable a) and 12 (from variable b) to the expression `a - b`.

```
bool d = a > b
```

Variables, values, and types



```
int a = 32
```

type variable value

```
int b = 12
```

```
int c = a - b
```

Diagram illustrating the calculation of `a - b`. Red arrows point from the value 32 (under `a`) and 12 (under `b`) to the result 20 (under `c`).

```
bool d = a > b
```

Variables, values, and types

a	b	c	d
32	12	20	true

int a = 32
type variable value

int b = 12

int c = a - b
32 12

bool d = a > b

Comments

```
int a = 32
// ignored
int b = 20
/*
ignored
int c = a-b
*/
bool d = a > b
```

Comments

```
int a = 32
```

```
// ignored
```

```
int b = 20
```

```
/*
```

```
ignored
```

```
int c = a-b
```

```
*/
```

```
bool d = a > b
```

The need for functions

```
int a = 2*2*2 - 2  
int b = 3*3*3 - 3  
int c = a-b
```

The need for functions

```
int a = 2*2*2 - 2
```

```
int b = 3*3*3 - 3
```

```
int c = a-b
```

```
int a = 2-ish magic
```

```
int b = 3-ish magic
```

```
int c = a-b
```

The need for functions

```
int a = 2*2*2 - 2  
int b = 3*3*3 - 3  
int c = a-b
```

```
int a = 2-ish magic  
int b = 3-ish magic  
int c = a-b
```

```
int magic (int x) {  
    return x*x*x - x  
}
```

```
int a = magic(2)  
int b = magic(3)  
int c = a-b
```

The need for functions

Tired

```
int a = 2*2*2 - 2
int b = 3*3*3 - 3
int c = a-b
```

Wired

```
int a = 2-ish magic
int b = 3-ish magic
int c = a-b
```

Inspired

```
int magic (int x) {
    return x*x*x - x
}
```

```
int a = magic(2)
int b = magic(3)
int c = a-b
```

Playing with functions

```
int magic (int x) {  
    return x*x*x - x  
}
```

```
int a = magic(2)  
int b = magic(3)  
int c = a-b
```

Playing with functions

```
int namemagic (int x) {  
    return x*x*x - x  
}
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```


Playing with functions

```
    return type name  
    int magic (int x) {  
        return x*x*x - x  
    }
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```

Playing with functions

```
    return type name argument  
    int magic (int x) {  
        return x*x*x - x  
    }
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```

Playing with functions

```
    return type    name    argument  
    int magic (int x) {  
        return x*x*x - x  
    }
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```

Playing with functions

```
    return type    name    argument  
    int magic (int x) { body starts  
        return x*x*x - x  
    }
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```

Playing with functions

```
    return type name argument  
    int magic (int x) { body starts  
        type name  
        return x*x*x - x  
  
    body ends }
```

```
int a = magic(2)
```

```
int b = magic(3)
```

```
int c = a-b
```

Playing with functions

```
    return type name argument  
    int magic (int x) { body starts  
        type name  
        return x*x*x - x  
  
    body ends }
```

```
int a = magic(2)  
int b = magic(3)  
int c = a-b
```

Playing with functions

```
    return type name argument  
    int magic (int x) { body starts  
        type name  
        return x*x*x - x  
  
    body ends }
```

```
int a = magic(2) 6  
int b = magic(3)  
int c = a-b
```

Playing with functions

```
    return type    name    argument
    int magic (int x) { body starts
                      type    name
                      return x*x*x - x
body ends }
```

```
int a = magic(2) → 6
int b = magic(3) →
int c = a-b
```


Playing with functions

```
    return type name argument  
    int magic (int x) { body starts  
        type name  
        return x*x*x - x  
  
    body ends }
```

```
int a = magic(2) 6  
int b = magic(3) 24  
int c = a-b
```

Fancier functions

```
int magic (int x) {  
    return x*x*x - x  
}
```

Fancier functions

```
int magic (int x) {  
    return x*x*x - x  
}
```

```
int bigger (int x, int y) {  
    if x > y  
    then return x  
    else return y  
}
```

Fancier functions

```
int magic (int x) {  
    return x*x*x - x  
}
```

```
int bigger (int x, int y) {  
    if x > y  
    then return x  
    else return y  
}
```

```
bool positive (int x) {  
    if x >= 0  
    then return true  
    else return false  
}
```

Fancier functions

```
int magic (int x) {  
    return x*x*x - x  
}
```

```
int bigger (int x, int y) {  
    if x > y  
    then return x  
    else return y  
}
```

```
bool positive (int x) {  
    if x >= 0  
    then return true  
    else return false  
}
```

```
bool combined () {  
    int a = magic(2)  
    int b = magic(-1)  
    int c = bigger(a,b)  
    return positive(c)  
}
```

Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

```
void sendAPacketTo(Router r) {  
    // creates and sends a packet to the router r  
}
```

Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

```
void sendAnamePacketTo(Router r) {  
    // creates and sends a packet to the router r  
}
```


Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

```
return type      name
void sendAPacketTo(Router r) {
    // creates and sends a packet to the router r
}
```

Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

```
return type      name      argument
void sendAPacketTo(Router r) {
    // creates and sends a packet to the router r
}
```

Fancier types

Later, you will see some strange types.

Not `int` and `bool`, but `Person`, `Letter`, `Router`, and `Packet`.

Try to remember how to break it down!

```
return type      name      argument
void sendAPacketTo (Router r) {
    // creates and sends a packet to the router r
}
```

That's all, have fun!

Don't worry if this felt a bit fast.

We'll provide a cheatsheet version of these slides...

But honestly, please just ask us :)