

COP 3223H: Introduction to C Programming

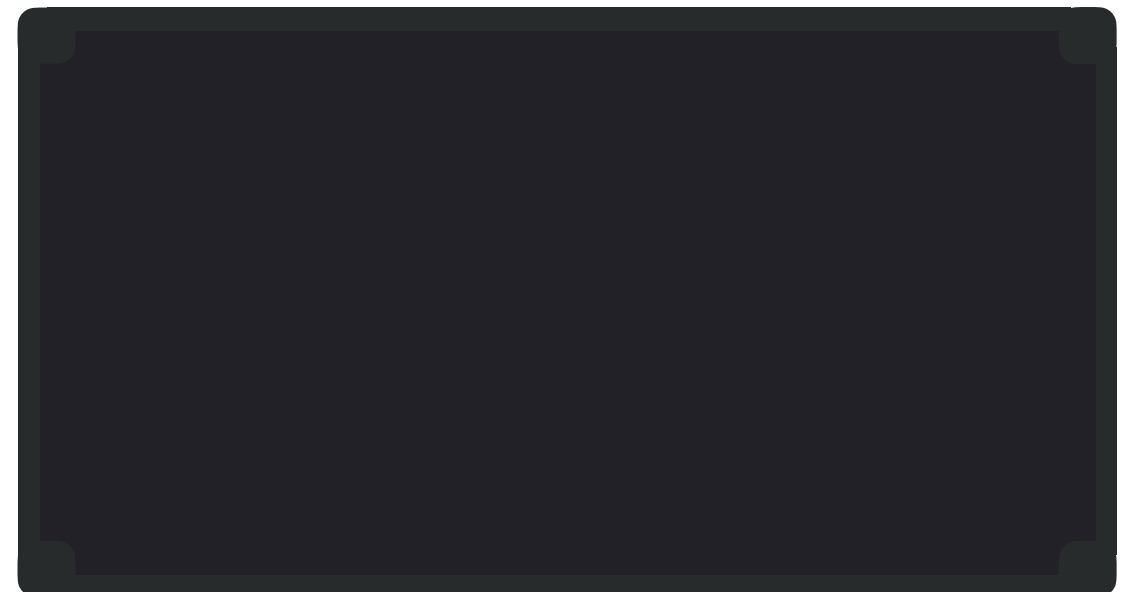
Fall 2023



University of
Central Florida

Dr. Kevin Moran

Class will start in:
10:00



COP 3223H: Introduction to C Programming

Fall 2023



University of
Central Florida

Dr. Kevin Moran

Week 4 - Class 2: Conditionals and *if* Statements





- *Small Programming Assignment 2* and *Large Programming Assignment 1* will come out early next week
 - I will be adjusting the timing of *Small Programming Assignment 3* - moving to later in the semester
- *Quiz 1* will be due Wednesday at 11:59 pm
 - Will be released today
- Heads up on *Exam 1*
 - Will be next Friday (September 22nd)
 - We will review extensively in class

Today's Agenda



1. Quick Review on User Defined Functions/Control Structures
2. Deep Dive on If Statements and Conditionals

Review of UDF and Control Structures



Types of Functions



- There are two types of functions.
 - Functions that *return a value*.
 - Functions that *don't return a value*.
- These types of functions are defined through their prototypes.
 - Functions that don't return a value have the reserved word `void` in front of the name.
 - Functions that do return a value have the type of data (`int`, `double`, `char`) in front of the of the name.

Functions with Arguments/Parameters



- Subproblems may require additional information in order to solve.
- Arguments send information over to functions.
- Input arguments are used to pass information into a function subprogram.
- Functions can have multiple arguments.

Functions with Arguments/Parameters



```
#include<stdio.h>
```

```
int mySecretFormula (int num, int num2, int num3);
```

```
int main ()  
{
```

```
    int num1 = 3;
```

```
    int num2 = 2;
```

```
    int num3 = 1;
```

```
int x = mySecretFormula (num1, num2, num3) ;
```

```
printf ("x = %d\n", x);
```

```
return 0;
```

```
}
```

```
int mySecretFormula (int num1, int num2, int num3)
```

```
{
```

```
int result = num1 + num2 * num3 - num3;
```

```
return result;
```

```
}
```

Parameters

Arguments

Recap of Last Class (cont.)



- Control structures are a combination of individual instructions into a single logical unit with one entry point and one exit point
- Compound Statement is a group of statements bracketed { and } that are executed sequentially.

```
int main(void)
{
    printf("Hello World \n");
    return 0;
}
```

```
int main(void)
{
    return 0;
}
```

If Statements



The If Statement



- Conditions are setup in the if statement.
- Syntax example

```
if(num1 < num2)
{
    printf("num1 is smaller than num2. \n");
}else
{
    printf("num2 is smaller than num1. \n");
}
```

Condition

Statement Executed if
Condition is "true"

Statement Executed if
Condition is "false"

If Statement Example # 1



Here



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	

If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
  Here → int num1;
         int num2;

  scanf("%d%d", &num1, &num2);

  if(num1 < num2)
  {
    printf("num1 is smaller than num2. \n");
  }else
  {
    printf("num2 is smaller than num1. \n");
  }

  return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	num1 = ???

If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
  Here → int num1;
          int num2;

  scanf("%d%d", &num1, &num2);

  if(num1 < num2)
  {
    printf("num1 is smaller than num2. \n");
  }else
  {
    printf("num2 is smaller than num1. \n");
  }

  return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = ???
AA0	num1 = ???

If Statement Example # 1



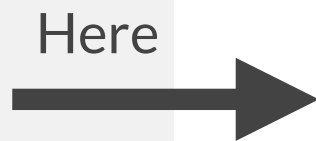
```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```



Let's collect
the values 4
and 7.

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 7
AA0	num1 = 4

If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    Here → if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

The condition is true!

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 7
AA0	num1 = 4

If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    Here → if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 7
AA0	num1 = 4

If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        Here → printf("num1 is smaller than num2. \n");
    } else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 7
AA0	num1 = 4

If Statement Example # 1



```
#include <stdio.h>

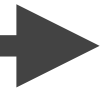
int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Here



Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 7
AA0	num1 = 4

If Statement Example # 2



Here



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
  Here → int num1;
         int num2;

  scanf("%d%d", &num1, &num2);

  if(num1 < num2)
  {
    printf("num1 is smaller than num2. \n");
  }else
  {
    printf("num2 is smaller than num1. \n");
  }

  return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	num1 = ??

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
  Here → int num1;
          int num2;

  scanf("%d%d", &num1, &num2);

  if(num1 < num2)
  {
    printf("num1 is smaller than num2. \n");
  }else
  {
    printf("num2 is smaller than num1. \n");
  }

  return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = ??
AA0	num1 = ??

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    Here → scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Let's collect
the values 14
and 5.

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 14

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    Here → if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 14

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    Here → if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

The condition is false!

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 14

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        Here → printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 14

If Statement Example # 2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }

    return 0;
}
```

Here



Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 14

If Statement with One Alternative



- Conditions are setup in the `if` statement.
- Syntax example

```
if(num1 != num2)  
    printf("num1 does not equal num2. \n");
```

Q&A: What happens if the condition is false?

- a) Program crashes at runtime
- b) Program does not execute the `printf` statement
- c) Program won't compile
- d) None of the above

Compound If Statement Example # 1



Here



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

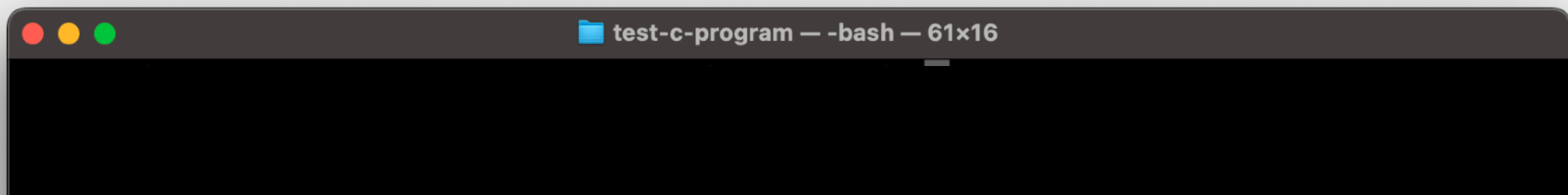
    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	



Compound If Statement Example # 1



```
#include <stdio.h>
```

```
int main(void)
```

Here

```
{
```

```
int num1;  
int num2;
```

```
scanf("%d%d", &num1, &num2);
```

```
if(num1 != num2)
```

```
{
```

```
printf("num1 is smaller than num2. \n");  
printf("Still in the true block. \n");
```

```
}else
```

```
{
```

```
printf("num2 is smaller than num1. \n");  
printf("Still in the false block. \n");
```

```
}
```

```
printf("I will always be displayed! \n");
```

```
return 0;
```

```
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	num1 = ??

test-c-program — -bash — 61x16

Compound If Statement Example # 1



Here



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

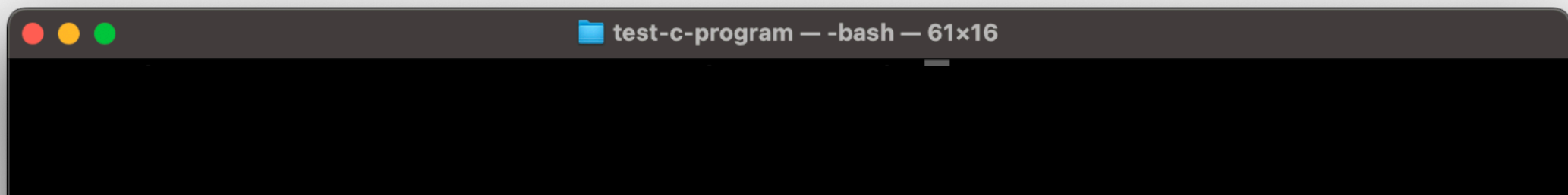
    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = ??
AA0	num1 = ??



Compound If Statement Example # 1



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

Here

```
    scanf("%d%d", &num1, &num2);
```

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

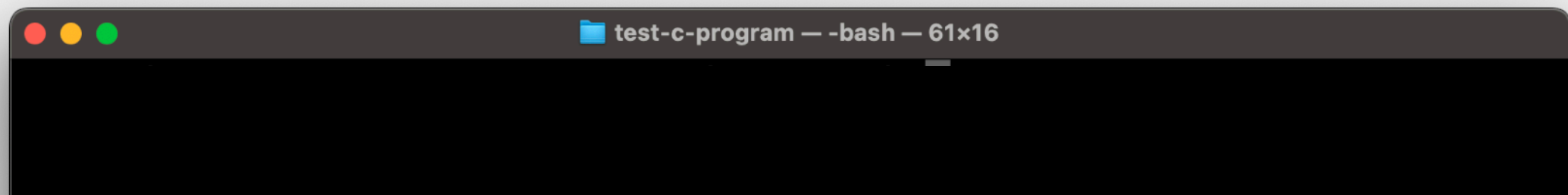
```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

Let's collect
the values 17
and 8.

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17



Compound If Statement Example # 1



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

```
    scanf("%d%d", &num1, &num2);
```

Here

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

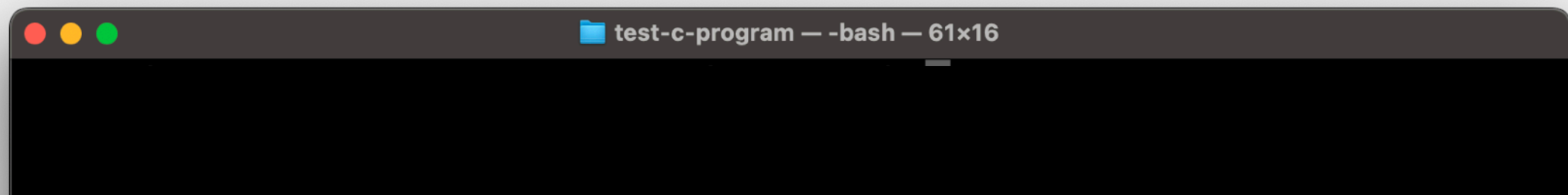
```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

The condition is true!

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17



Compound If Statement Example # 1



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

```
    scanf("%d%d", &num1, &num2);
```

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

The condition is true!

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program --bash -- 61x16
num1 is smaller than num2.
```

Compound If Statement Example # 1



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

```
    scanf("%d%d", &num1, &num2);
```

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

The condition is true!

Here



Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program --bash -- 61x16
num1 is smaller than num2.
Still in the true block.
```

Compound If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

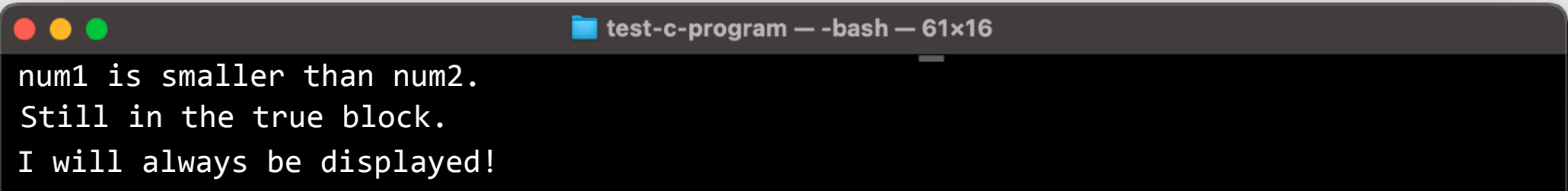
    printf("I will always be displayed! \n");

    return 0;
}
```

The condition is true!

Here →

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17



Compound If Statement Example # 1



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

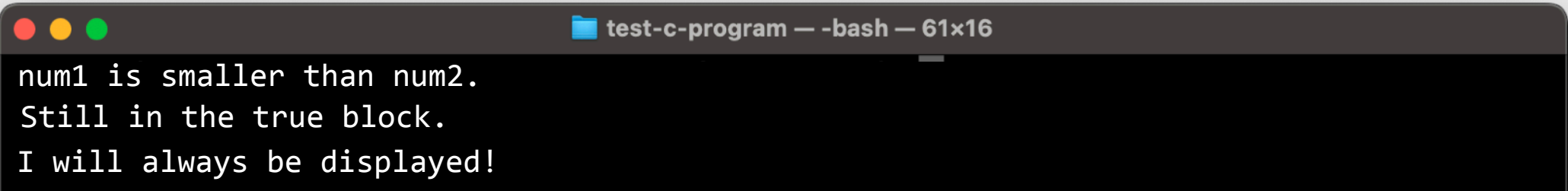
    printf("I will always be displayed! \n");

    return 0;
}
```

The condition is true!

Here →

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17



Compound If Statement Example #2



Here



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

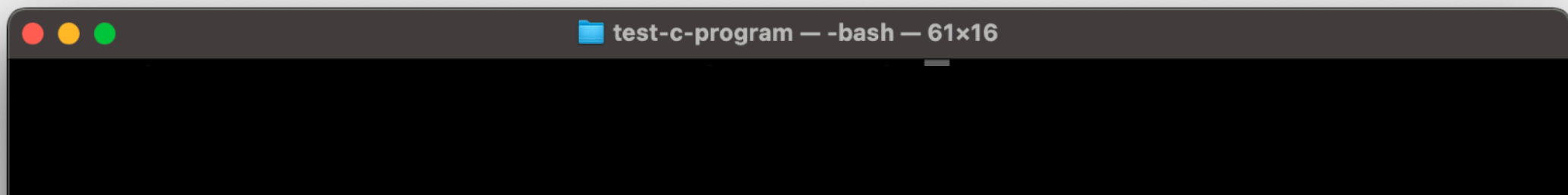
    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	



Compound If Statement Example #2



```
#include <stdio.h>
```

```
int main(void)
```

Here

```
{
```

```
int num1;  
int num2;
```

```
scanf("%d%d", &num1, &num2);
```

```
if(num1 != num2)
```

```
{
```

```
printf("num1 is smaller than num2. \n");  
printf("Still in the true block. \n");
```

```
}else
```

```
{
```

```
printf("num2 is smaller than num1. \n");  
printf("Still in the false block. \n");
```

```
}
```

```
printf("I will always be displayed! \n");
```

```
return 0;
```

```
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	
AA0	num1 = ??

test-c-program — -bash — 61x16

Compound If Statement Example #2



```
#include <stdio.h>
```

```
int main(void)  
{
```

Here

```
    int num1;  
    int num2;
```

```
    scanf("%d%d", &num1, &num2);
```

```
    if(num1 != num2)  
    {
```

```
        printf("num1 is smaller than num2. \n");  
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {  
        printf("num2 is smaller than num1. \n");  
        printf("Still in the false block. \n");
```

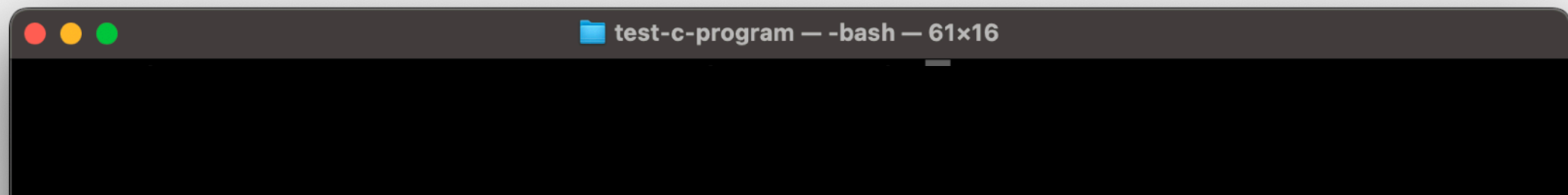
```
    }
```

```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = ??
AA0	num1 = ??



Compound If Statement Example #2



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

Here

```
    scanf("%d%d", &num1, &num2);
```

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

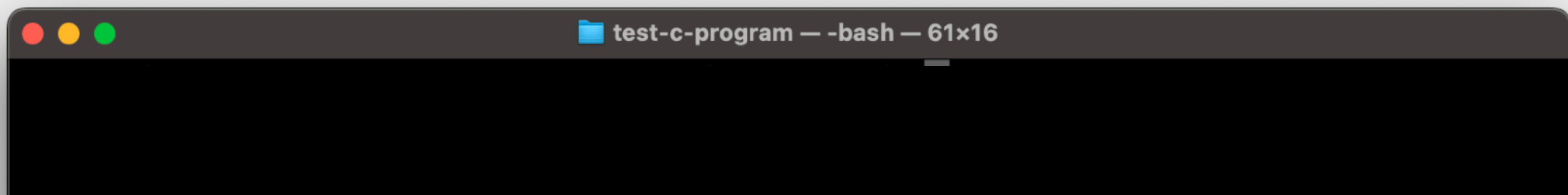
```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

Let's collect
the values 5
and 5.

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 5



Compound If Statement Example #2



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1;
```

```
    int num2;
```

```
    scanf("%d%d", &num1, &num2);
```

Here

```
    if(num1 != num2)
```

```
    {
```

```
        printf("num1 is smaller than num2. \n");
```

```
        printf("Still in the true block. \n");
```

```
    }else
```

```
    {
```

```
        printf("num2 is smaller than num1. \n");
```

```
        printf("Still in the false block. \n");
```

```
    }
```

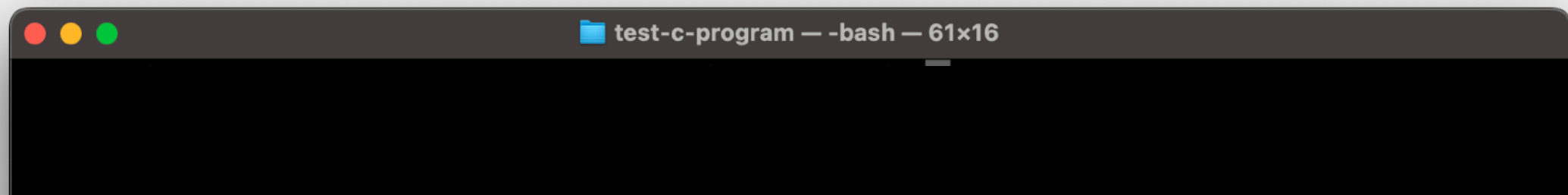
```
    printf("I will always be displayed! \n");
```

```
    return 0;
```

```
}
```

The condition is false!

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 5
AA0	num1 = 5



Compound If Statement Example #2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        Here → printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program --bash -- 61x16
num2 is smaller than num1.
```

Compound If Statement Example #2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        Here → printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program — -bash — 61x16
num2 is smaller than num1.
Still in the false block.
```

Compound If Statement Example #2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Here



Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program --bash -- 61x16
num2 is smaller than num1.
Still in the false block.
I will always be displayed!
```

Compound If Statement Example #2



```
#include <stdio.h>

int main(void)
{
    int num1;
    int num2;

    scanf("%d%d", &num1, &num2);

    if(num1 != num2)
    {
        printf("num1 is smaller than num2. \n");
        printf("Still in the true block. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
        printf("Still in the false block. \n");
    }

    printf("I will always be displayed! \n");

    return 0;
}
```

Here



Stack	Space
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	
AA2	
AA1	num2 = 8
AA0	num1 = 17

```
test-c-program --bash -- 61x16
num2 is smaller than num1.
Still in the false block.
I will always be displayed!
```



Guess What!! You can include control structures if you have multiple statements that need to be properly executed!!

e”

Multiple Alternative `if-else` Example



- Suppose you want to associate noise loudness measured in decibels with the effect of the noise. The following table shows the relationship between noise levels and human perceptions of noises. How can this be expressed with multiple-alternative if-else statements?

Loudness in Decibels (db)	Perception
50 or lower	quiet
51-70	intrusive
71-90	annoying
91-110	very annoying
Above 110	uncomfortable

Nested `if` Statements



- After testing and determining the outcome, it is possible to dive into another condition.
- This is known as creating nested statements.
- Think about nesting dolls!
- Inside a nest doll is another doll. Inside a nest if statement is another if statement.

```
if (num1 != 0)
    if(num1 !=1)
        if(num1!=2)
            if(num1!=3)
                printf("num is neither 0, 1, 2, or 3 ...");
```

switch Statements



switch Statement



- Some of the `if else` statements can deal with checking for an exact match.
- What would happen if there are lots of multiple-alternative `if-else` statements that dealt with only equality checks
- Switch Statement allows programmers to write a cleaner version of `if-else` that only deals with `==` operator.

Q&A: Switch statements use relational operators for comparison?

a) True

b) False

switch Statement Syntax



```
switch(ticket) ← variable being evaluated for equality
{
  case 1: ← ticket == 1
    printf("Proceed to entrance 1.\n");
    break;

  case 2: ← ticket ==2
    printf("Proceed to entrance 2.\n");
    break;

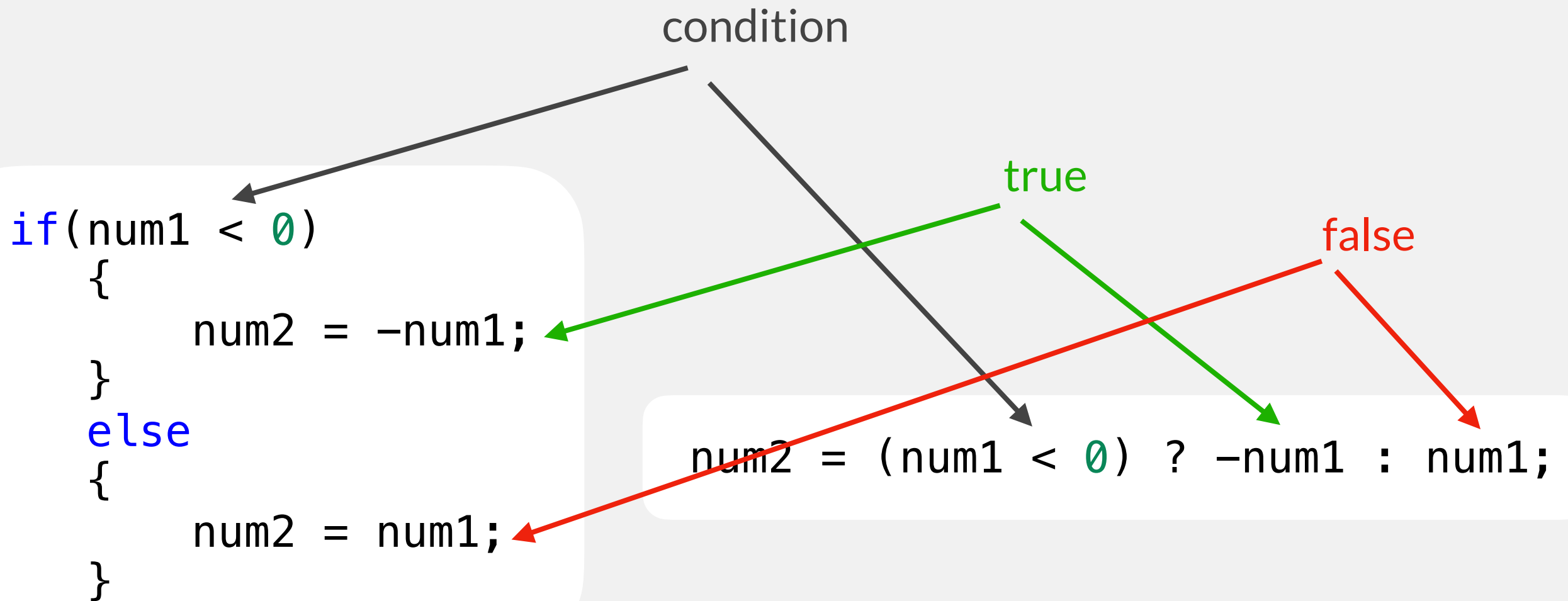
  case 3: ← ticket ==3
    printf("Proceed to entrance 3.\n");
    break;

  default: ← ticket !=1 && ticket !=2 && ticket !=3
    printf("Sorry, your ticket does not match!");
}
}
```

The Conditional Operator ? :



- C offers a quick simple way to write out an if-else statement in one line of code.





Slides adapted from Dr. Andrew Steinberg's
COP 3223H course