A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Programming Techniques COSC1284/2010

Tutorial 3



Agenda

- Tutorial/Lab
 - Read chapter 3 from the textbook
 - Read chapter 4 from the textbook
 - Discuss the concepts with your tutor and fellow classmates
 - Complete chapter 3 - Exercises 1 - 2
 - Complete chapter 4 - Exercises 1 - 3
 - Attempt on your own
 - Complete chapter 3 - Exercises 3 - 4
 - Complete chapter 4 - Exercises 4 - 6



Exercise 3.1

- When you use `printf`, the `Java` compiler does not check your format string. See what happens if you try to display a value with type `int` using `%f`. And what happens if you display a `double` using `%d`? What if you use two format specifiers, but then only provide one value?



Exercise 3.2

- Write a program that converts a temperature from Celsius to Fahrenheit. It should
 1. Prompt the user for input
 2. Read a double value from the keyboard
 3. Calculate the result
 4. Format the output to one decimal place. For example, it should display "24.0 C = 75.2 F".
- Here is the formula. Be careful not to use integer division!

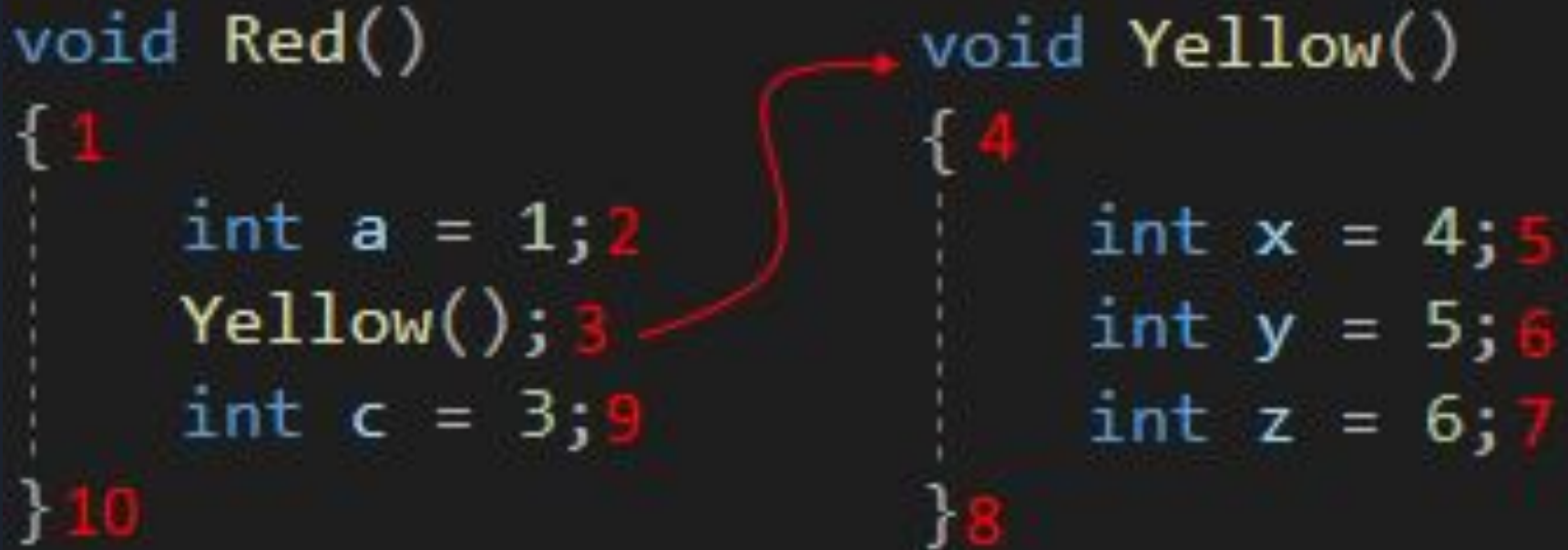
$$F = C \times \frac{9}{5} + 32$$



Methods

- Writing all your code in the main method is a bad idea, we come to a point where we have to break up the code into multiple methods.
- Today's exercises will show the following:
 - Show you how to organize programs into multiple methods.
 - Learn how to trace the ordering in which a program runs.
 - Discuss strategies for incrementally developing and testing your code.
- Also we will be talking about parameters and return types.
- Note: A method (in general) should represent a single task, such as walk, login, close, etc.

Program Flow (Multiple Methods)



```
void Red()  
{  
    1  
    int a = 1; 2  
    Yellow(); 3  
    int c = 3; 9  
} 10  
  
void Yellow()  
{  
    4  
    int x = 4; 5  
    int y = 5; 6  
    int z = 6; 7  
} 8
```

Passing and Returning Variables between Methods

```
void Red()  
{  
    int a = 1;  
    int b = 2;  
    int c = Yellow(a, b);  
    print(c);  
}  
  
int Yellow(int a, int b)  
{  
    return a + b;  
}
```

The diagram illustrates the flow of data between two methods. A red arrow originates from the `Yellow(a, b)` call in the `Red()` method and points to the `Yellow` method definition, indicating the call. A yellow arrow originates from the `return a + b;` statement in the `Yellow` method and points back to the `int c =` assignment in the `Red()` method, showing the return value being passed back to the caller.



Exercise 4.1

- The purpose of this exercise is to take code from a previous exercise and redesign it as a method that takes parameters. You should start with a working solution to Exercise 2.
- Write a `method` called `printAmerican` that takes the `day`, `date`, `month` and `year` as parameters and that displays them in American format.
- Test your `method` by invoking it from main and passing appropriate arguments. The output should look something like this (except that the date might be different):
 - Saturday, July 22, 2015
- Once you have debugged `printAmerican`, write another method called `printEuropean` that displays the date in European format.

Exercise 4.2

- This exercise reviews the flow of execution through a program with multiple methods. Read the following code and answer the questions.

```
public static void main(String[] args) {  
    zippo("rattle", 13);  
}
```

```
public static void baffle(String blimp) {  
    System.out.println(blimp);  
    zippo("ping", -5);  
}
```

```
public static void zippo(String quince, int flag) {  
    if (flag < 0) {  
        System.out.println(quince + " zoop");  
    } else {  
        System.out.println("ik");  
        baffle(quince);  
        System.out.println("boo-wa-ha-ha");  
    }  
}
```

- What is the value of the parameter blimp when baffle (the method) gets invoked?
- What is the output of this program?



Exercise 4.3

```
public static void zoop() {  
    baffle();  
    System.out.print("You wugga ");  
    baffle();  
}
```

```
public static void main(String[] args) {  
    System.out.print("No, I ");  
    zoop();  
    System.out.print("I ");  
    baffle();  
}
```

```
public static void baffle() {  
    System.out.print("wug");  
    ping();  
}
```

```
public static void ping() {  
    System.out.println(".");  
}
```

- Draw a stack diagram that shows the state of the program the first time ping is invoked.
- What is output by the following program? Be precise about where there are spaces and where there are newlines.