# Java Programming Foundations 1

## Week 4: Beginning Control Flow

- Introduction to control flow and conditionals.
- The `if` statement.
- `if else` and `else`.
- The `switch` statement.

# What is control flow?

# Control flow is the order in which statements are executed

# What are conditionals?

# Conditionals determine which path your code will take

# Run code only when a condition is met

# Think "if this, then that"

# Control flow and conditionals in everyday language

## Example: you're at your favorite restaurant about to order.

***If*** they have the fish, ***then*** I will order fish.

## Example: you're at your favorite restaurant about to order.

- ***If*** they have the fish, ***then*** I will order fish.
- `they have the fish` is the condition,
- `I will order fish` is the result when the condition is met.

## Example: you're driving on the highway where the speed limit is 70 miles per hour.

***If*** I drive over 70 miles per hour, ***then*** I will get a ticket.

## Example: you're driving on the highway where the speed limit is 70 miles per hour.

- ***If*** I drive over 70 miles per hour, ***then*** I will get a ticket.
- `I drive over 70 miles per hour` is the condition,
- `I will get a ticket` is the result when the condition is met.

## Example: you want to buy orange juice

***If*** I have at least $2.75, ***then*** I will buy an orange juice.

## Example: you want to buy orange juice

- ***If*** I have at least $2.75, ***then*** I will buy an orange juice.
- `I have at least $2.75` is the condition,
- `I will buy an orange juice` is the result when the condition is met.

# Control flow in Java

# The `if` statement.

# Example

```
if (yourHeightInFeet >= 5) {
  System.out.println("You are allowed to enter the ride.");
}
```

Use an `if` statement when you want code to run only when a condition is met.

In Java, statements like `if` are referred to as conditionals

# The `if` statement's syntax

```
// Code before the if statement, this always runs.

if (/* condition */) {
  // Code to run when `condition` is true.
  // ...
}

// Code after the if statement, this always runs.
```

# What can I put as my `condition`?

```
// Code before the if statement, this always runs.

if (/* condition */) {
  // Code to run when `condition` is true.
  // ...
}

// Code after the if statement, this always runs.
```

The `condition` can be any `boolean` expression

The `condition` can be anything that evaluates to `true` or `false`

`boolean` values are either `true` or `false`

# Boolean operators: `boolean` values

- `a && b`, the and operator, both `a` and `b` are `true`.
- `a || b`, the or operator, either `a` or `b` are `true`.
- `!a`, the not operator, the inverse (opposite) of `a`.

# Boolean logic exercises: `boolean` values

1. `true && true`
2. `true && false`
3. `false && true`
4. `false && false`
5. `true || true`
6. `true || false`
7. `false || true`
8. `false || false`
9. `!true`
10. `!false`

# Comparison operators: `char`, `int`, and `double` values

- `a == b`, `a` is equal to `b`.
- `a != b`, `a` is not equal to `b`.
- `a > b`, `a` is greater than `b`.
- `a >= b`, `a` is greater than or equal to `b`.
- `a < b`, `a` is less than `b`.
- `a <= b`, `a` is less than or equal to `b`.

# Comparison operator exercises: `char`, `int`, and `double` values

1. `42 == 42`
2. `42 != 42`
3. `84 > 32`
4. `84 < 32`
5. `43 <= 32`
6. `54 <= 54`
7. `12 <= 13`

# Reminder: `if` statement's syntax

```
// Code before the if statement, this always runs.

if (/* condition */) {
  // Code to run when `condition` is true.
  // ...
}

// Code after the if statement, this always runs.
```

Let's write some code: if I have at least $2.75, then I will buy an orange juice.

```java
public class Sample01 {
  // ***If*** I have at least $2.75, ***then*** I will buy an orange juice.
  public static void main(String[] args) {
    double money = 5.65;

    System.out.println("You have " + money);

    if (money >= 2.75) {
      System.out.println("You will buy orange juice");
    }

    System.out.println("Done running program.");
  }
}
```

```java
import java.util.Scanner;

public class Sample02 {
  // ***If*** I have at least $2.75, ***then*** I will buy an orange juice.
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    System.out.print("Please enter how much money you have: ");
    double money = input.nextDouble();

    System.out.println("You have " + money);

    if (money >= 2.75) {
      System.out.println("You will buy orange juice");
    }

    System.out.println("Done running program.");
  }
}
```

## Example: you want to buy orange juice

***If*** I have at least $2.75 and the store sells orange juice, ***then*** I will buy an orange juice.

## Example: you want to buy orange juice

- ***If*** I have at least $2.75 and the store sells orange juice, ***then*** I will buy an orange juice.
- `I have at least $2.75 and the store sells orange juice` is the condition,
    - There are two parts to this condition and both must be true in order to proceed:
        a. `I have at least $2.75` must be true
        b. `the store sells orange juice` must be true
- `I will buy an orange juice` is the result when the condition is met.

# Let's write some code: update previous example to use new condition

# `if` statements: the `else` clause

# Think "if this is true, then do A, otherwise do B"

## Example: you're at your favorite restaurant about to order.

***If*** they have the fish, ***then*** I will get the fish, ***otherwise*** I will get a salad.

## Example: you're at your favorite restaurant about to order.

- ***If*** they have the fish, ***then*** I will get the fish, ***otherwise*** I will get a salad.
- `they have the fish` is the condition,
- `I will get the fish` is the result when the condition is met,
- `I will get a salad` is the result when the condition is not met.

# Syntax of an `if` statement with an `else` clause

```
// Code before the if statement, this always runs.

if (/* condition */) {
  // Code to run when `condition` is true.
  // ...
} else {
  // Code to run when `condition` is not true.
  // ...
}

// Code after the if statement, this always runs.
```

# Let's write some code

```java
public class Sample03 {
  // ***If*** they have the fish,
  // ***then*** I will get the fish,
  // ***otherwise*** I will get a salad.
  public static void main(String[] args) {
    boolean fishIsAvailable = true;

    if (fishIsAvailable) {
      System.out.println("I will get the fish");
    } else {
      System.out.println("I will get a salad");
    }

    System.out.println("Done running program.")
  }
}
```

# `if` statements: the `else if` constructor

Use an `if` statement with an `else if` constructor when you are choosing between multiple options all of which have conditions.

## Example: you want either orange juice, or a smoothie, or water depending on what's available.

***If*** the store has orange juice, ***then*** I will get orange juice, ***otherwise if*** the store has smoothies, ***then*** I will get a smoothie, ***otherwise if*** the store has water, ***then*** I will get water.

## Example: you want either orange juice, or a smoothie, or water depending on what's available.

- ***If*** the store has orange juice, ***then*** I will get orange juice, ***otherwise if*** the store has smoothies, ***then*** I will get a smoothie, ***otherwise if*** the store has water, ***then*** I will get water.
- `the store has orange juice` is the first condition,
- `I will get orange juice` is the result when the first condition is met,
- `the store has smoothies` is the second condition,
- `I will get a smoothie` is the result when the second condition is met,
- `the store has water` is the third condition,
- `I will get water` is the result when the third condition is met.

# Order of conditions matters

# Syntax of an `if` statement with an `else if` constructor

```
if (/* conditionA */) {
  // Code to run when `conditionA` is true.
  // ...
} else if (/* conditionB */) {
  // Code to run when `conditionB` is true.
  // ...
}
```

Let's write some code: if the store has orange juice, then I will get orange juice, otherwise if the store has smoothies, then I will get a smoothie, otherwise if the store has water, then I will get water.

```java
public class Sample04 {
  // ***If*** the store has orange juice, ***then*** I will get orange juice,
  // ***otherwise if*** the store has smoothies, ***then*** I will get a
  // smoothie, ***otherwise if*** the store has water, ***then*** I will get
  // water.
  public static void main(String[] args) {
    boolean hasOrangeJuice = true;
    boolean hasSmoothies = true;
    boolean hasWater = true;

    if (hasOrangeJuice) {
      System.out.println("I will get orange juice");
    } else if (hasSmoothies) {
      System.out.println("I will get a smoothie");
    } else if (hasWater) {
      System.out.println("I will get water");
    }
  }
}
```

What happens if we don't use `else if` and only use `if` statements?

# When to use what?

- Use an `if` statement when you want code to run only when a condition is met.
- Use an `if` statement with an `else` clause when you are choosing between an option with a condition and a default option.
- Use an `if` statement with an `else if` constructor when you are choosing between multiple options all of which have conditions.

# Putting it all together

# Putting it all together

```
if (/* conditionA */) {
  // Code to run when `conditionA` is true.
  // ...
} else if (/* conditionB */) {
  // Code to run when `conditionB` is true.
  // ...
} else {
  // Code to run when none of the previous conditions are true.
  // ...
}
```

# Putting it all together

```
if (/* conditionA */) {
  // Code to run when `conditionA` is true.
  // ...
} else if (/* conditionB */) {
  // Code to run when `conditionB` is true.
  // ...
} else if (/* conditionC */) {
  // Code to run when `conditionC` is true.
  // ...
} else {
  // Code to run when none of the previous conditions are true.
  // ...
}
```

## `if` statements must start with an `if`

# The `else` is optional

# Let's write some code: convert percentage grade to letter grade

```java
import java.util.Scanner;

public class Sample06 {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    System.out.print("What is your grade? ");
    double grade = input.nextDouble();

    if (grade < 0 || grade > 100) {
      System.out.println("You grade must be between 0 to 100");
    } else if (grade >= 90) {
      System.out.println("You get an A");
    } else if (grade >= 80) {
      System.out.println("You get a B");
    } else if (grade >= 70) {
      System.out.println("You get a C");
    } else {
      System.out.println("You get an F");
    }
  }
}
```

Note that you can have multiple statements inside of the curly brackets.

# Nested conditionals

## `if` statements inside of `if` statements

```
if (/* conditionA */) {
  // This code runs when `conditionA` is true.

  if (/* conditionB */) {
    // This code runs when `conditionA` and `conditionB` are both true.
  }

  // This code runs when `conditionA` is true.
}
```

Let's write some code: if the store is open then buy juice, or smoothie, or water.

```java
public class Sample05 {
  public static void main(String[] args) {
    boolean storeIsOpen = true;

    boolean hasOrangeJuice = true;
    boolean hasSmoothies = true;
    boolean hasWater = true;

    if (storeIsOpen) {
      if (hasOrangeJuice) {
        System.out.println("I will get orange juice");
      } else if (hasSmoothies) {
        System.out.println("I will get a smoothie");
      } else if (hasWater) {
        System.out.println("I will get water");
      }
    } else {
      System.out.println("The store is closed");
    }
  }
}
```

# A note on formatting your code.

For easier to read code, pay attention to your indentation.

# A note on the syntax of `if` statements

## The curly brackets are optional but you are encouraged to always include them.

```
if (hasOrangeJuice)
  System.out.println("I will get orange juice");
```

# What could go wrong if I exclude curly brackets?

# Let's write some *bad* code

```
public class Sample05 {
  public static void main(String[] args) {
    boolean storeIsOpen = true;

    boolean hasOrangeJuice = false;
    boolean hasSmoothies = false;
    boolean hasWater = false;

    if (storeIsOpen)
      if (hasOrangeJuice)
        System.out.println("I will get orange juice");
      else if (hasSmoothies)
        System.out.println("I will get a smoothie");
      else if (hasWater)
        System.out.println("I will get water");
    else
      System.out.println("The store is closed");
  }
}
```

# Common mistakes with `if` statements

- Multiple statements without curly brackets.
- Dangling `else`.
- Misplaced semicolons.

# The `switch` statement.

`switch` statements are similar to `if` statements.

# The `switch` statement's syntax

```
// Code before the switch statement, this always runs.

switch (/* expression */) {
  case /* comparisonA */:
    // Code to run when `expression` is equal to `comparisonA`
    // ...
    break;

  case /* comparisonB */:
    // Code to run when `expression` is equal to `comparisonB`
    // ...
    break;

  default:
    // Code to run when none of the previous conditions were met.
    // ...
    break;
}

// Code after the switch statement, this always runs.
```

# Let's look at an example

```java
import java.util.Scanner;

public class Sample06 {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    System.out.println("How would you like to travel?");
    System.out.print("Enter an option between 1 and 3: ");
    int option = input.nextInt();

    switch (option) {
      case 1:
        System.out.println("Option 1 gets you a car");
        break;

      case 2:
        System.out.println("Option 2 gets you a bike");
        break;

      case 3:
        System.out.println("Option 3 gets you roller blades");
        break;

      default:
        System.out.println("You did not enter a valid option");
        break;
    }
  }
}
```

# How does `case` work?

```
int option = input.nextInt();

switch (option) {
  case 1:
    System.out.println("Option 1 gets you a car");
    break;

  case 2:
    System.out.println("Option 2 gets you a bike");
    break;

  case 3:
    System.out.println("Option 3 gets you roller blades");
    break;

  default:
    System.out.println("You did not enter a valid option");
    break;
}
```

# How does `switch` and `case` compare to an `if` statement?

## Using a `switch` statement

```
switch (option) {
  case 1:
    System.out.println("Option 1 gets you a car");
    break;
}
```

## Using an `if` statement

```
if (option == 1) {
  System.out.println("Option 1 gets you a car");
}
```

## What does does `break` do?

```
switch (option) {
  case 1:
    System.out.println("Option 1 gets you a car");
    break;

  case 2:
    System.out.println("Option 2 gets you a bike");
    break;

  case 3:
    System.out.println("Option 3 gets you roller blades");
    break;

  default:
    System.out.println("You did not enter a valid option");
    break;
}
```

## What does does `default` do?

```
switch (option) {
  case 1:
    System.out.println("Option 1 gets you a car");
    break;

  case 2:
    System.out.println("Option 2 gets you a bike");
    break;

  case 3:
    System.out.println("Option 3 gets you roller blades");
    break;

  default:
    System.out.println("You did not enter a valid option");
    break;
}
```

## What happens if we don't include the `break` statement?

## Common mistakes

- Missing `break` statement.

# Hint: how to read a character (a single letter) from user input

```
Scanner input = new Scanner(System.in);

char character = input.next().charAt(0);
```

# Hint: comparing `char` values

```
char myLetter = 'a';

if (myLetter == 'a') {
  // ...
} else if (myLetter == 'b') {
  // ...
} else {
  // ...
}
```