

The seal of Humberford Community College is a circular emblem. It features a central shield with a diagonal line. Above the shield is a banner with the text "HUMBERFORD COMMUNITY COLLEGE". Below the shield is the year "1938". The shield itself contains a torch and a book.

Decisions I

HFCC – CIS222

30 January 2012

Announcements:

✓ Quiz 3 on Wednesday (1 Feb)

- On User Input & Forms

✓ Homework 2 due 6 Feb (next Monday)

✓ Project Part 1 due 29 Feb

What is a Decision

- A choice based on a true/false condition
 - A **condition statement** is evaluated
 - The program does the code segment if the statement is true
 - Format:

```
if (condition statement) {  
    // Stuff to do when condition is true  
}
```

Condition Statements

➤ Must be unambiguous

- The value of the statement is either TRUE or FALSE
 - You cleaned your room
 - Your age is less than 13
 - The pizza has vegetables on it
- Often done through comparison
 - The grade is greater than or equal to 68
 - The color of the object is blue
 - The variable is empty

Example Decisions

```
if (your grade is greater than or equal to 68%) {
```

```
    // You pass this course
```

```
    echo "Pass";
```

```
}
```

```
if (your account has $20 or more) {
```

```
    // You may deduct money at the ATM
```

```
    echo "You may deduct money";
```

```
}
```

```
if (you are good at basketball) {
```

```
    // You may play in the NBA
```

```
    $playNBA = true;
```

```
}
```

Comparison Conditions

➤ Compares two values using **comparison operators**

- == equal to
(do the two sides have the same value?)
- < less than
(is the item on the left less than that on the right?)
- <= less than or equal to
(is the left side less than or equal to the right?)
- > greater than
(is the item on the left greater than that on the right?)
- >= greater than or equal to
(is the left side greater than or equal to the right?)
- != not equal to
(do the two sides have different values?)
- === exactly equal to
(are the two sides the same and the same data type?)

Condition Statements

```
if ($grade >= .68) {  
    // You pass this course  
    echo "Pass";  
}
```

```
if ($account >= 20) {  
    // You may deduct money at the ATM  
    echo "You may deduct money";  
}
```

```
if ($pagename == "") {  
    // Include the main page  
    $filename = "home.inc";  
}
```

Questions?



Let's Try It!

- Test some decision statements
- Create a basic page *InClass-DecisionBasics.php*
 - Set up a variable *\$price* to fetch its value from “*price*” in GET (*\$_GET['price']*)
 - Set up a decision that will output “Too much” if the price is over \$2.00
 - Upload and test with multiple values
- Raise your hand if you have finished (or if you are stuck)

Compound Conditions

➤ Merges two comparisons

- The whole condition must be true for the statement to be true
- Conditions are joined by operators for AND and OR
 - **AND uses &&** - both statements must be true for the condition to be true
 - **OR uses ||** - one or more statements must be true for the condition to be true
- Examples:

`if($num > 12 && $num < 20)     // Number in the teens`

`if($grade >= 84 && $grade < 92) // Letter grade B`

`if($time < 8 || $time > 17) // Time is outside of work hours`

Questions?



Let's Try It!

if(\$a < \$b && \$a < \$c)

F

\$a = 7;
\$b = 3;
\$c = 10;

if(\$a < \$b || \$a < \$c)

T

if(\$a + \$b == \$c && \$c < 20)

T

if(\$a == \$b && \$a < \$c)

T

If...Then...Else

➤ **Else** executes code when the true side doesn't execute

```
if($grade >= 68) {  
    // Happens if the value of $grade is 68 or higher  
    echo "Pass!";  
} else {  
    // Happens if the above block doesn't happen  
    echo "Fail.";  
}
```

Questions?



Let's Try It!

- Test some decision statements
- Create a basic page *InClass-DecisionElse.php*
 - Set up a variable *\$price* to fetch its value from “*price*” in GET (*\$_GET['price']*)
 - Set up a decision that will output “Too much” if the price is over \$2.00
 - The same decision should also output “Just right” if the price is under \$2.00
 - Upload and test with multiple values
- Raise your hand if you have finished (or if you are stuck)

If ... Else If ... Else

➤ **Else If** handles multiple affirmative cases

- For questions that have a range of responses
- Each **Else If** must have a condition to evaluate
- **Else** is still used to catch all other cases
- Only one case can happen in a single pass

```
if($grade >= 92) {  
    echo "A";  
} else if($grade >= 84) {  
    echo "B";  
} else if($grade >= 76) {  
    echo "C";  
} else if($grade >= 68) {  
    echo "D";  
} else {  
    echo "E";  
}
```

Questions?



Let's Try It!

- Test some decision statements
- Create a basic page *InClass-DecisionElself.php*
 - Set up a variable *\$price* to fetch its value from “*price*” in GET (*\$_GET['price']*)
 - Set up a decision that will output “Too much” if the price is over \$2.00
 - The same decision should also output “Just right” if the price is under \$2.00
 - That decision should also output “Error!” if the price is under \$0.00
 - Upload and test with multiple values
- Raise your hand if you have finished (or if you are stuck)

➤ **Switch** also handles multiple affirmative cases

- **Default** is used to catch all other cases (or to add a 'always on' case)

```
switch($menu) {  
    case 1:  
        echo "Checking Balance:";  
        break;  
    case 2:  
        echo "Withdrawing Funds:";  
        break;  
    case 3:  
        echo "Depositing Funds:";  
        break;  
    default:  
        echo "Exiting...";  
}
```

- **Break** is used to signal not to perform any other cases

Switch Cascading

- Unlike if blocks, in a switch, more than one case can be executed

```
switch($money) {  
    case ($money >= 20):  
        $tokens += $money;  
    case ($money >= 10):  
        $tokens += $money;  
    case ($money >= 5):  
        $tokens += $money;  
    default:  
        $tokens += ($money * 4);  
}
```

$\$m = 5;$

X

X

$\$t = 5$

$\$t = 25$

Questions?



Let's Try It!

- Use a switch statement
- Create a basic page *InClass-DecisionSwitch.php*
 - Set up a variable *\$num* to fetch its value from “*num*” in GET (*\$_GET['num']*) to collect a number from 1 to 10
 - Use a **switch** to do the following:
 - If the number is prime (1, 2, 3, 5, 7), output “Prime”
 - If the number is not prime, output “Not Prime”
 - If the number is not from 1 to 10, output “Out of Range”
 - Upload and test with multiple values
- **Raise your hand if you have finished (or if you are stuck)**

Next Session

➤ Loops and Repetition

- (Read CH2, pp95-105)