CMPSC 100

Computational Expression

Schedule update

Our course schedule reflects the structure of the rest of the semester.

The following slides describe changes made.

Schedule update

Office hours now take place on:

    Monday   9a - 11a

    Tuesday   9a - 11a

    Thursday 9a - 11a

    Friday   9a - 11a

These will be conducted using a standing Google Meet appointment.

Schedule update

Office hours will also now have specific purposes:

Monday	New topic introduction
Tuesday	Discussion of Practicals
Thursday	Discussion of Labs/Projects
Friday	Open office hours (any topic)

These will be conducted using a standing Google Meet appointment.

Schedule update

The only exception to this schedule will be *this week* where:

Wednesday	You will receive a narrated presentation
Thursday	Discussion of Practical 8
Friday	Discussion of Lab 9

These will be conducted using a standing [Google Meet appointment](#).

Schedule update

This [Google Meet appointment](#) is only for members of our CMPSC 100 course, but is always *for all members*, and will be recorded so that students can benefit from questions and conversation.

If you would prefer to have a private discussion, we need to set aside a different time by appointment. Please direct message me on our course Slack or email.

Schedule update

Monday office hour sessions will always consist of:
     1 hour of topic presentation and intro
      Q & A

I will update the links for Monday content with a
recording of this presentation.

Schedule update

Every presentation will be accompanied by a link at which viewers can ask questions (anonymously, if you prefer).

Keep this link open! You can up-vote questions you're interested in so that we can make our conversations more valuable. (The link will be different each time.)

Schedule update

Labs and Practicals will be assigned at the beginning of the week using the #labs and #practicals channels in our course Slack.

Each is due 1.5 weeks after assignment.

Strategies for success in a remote world

What I would do:

- Keep taking notes during conversations and presentations

- Work with others to attempt assignments

What I wouldn't do:

- Stop taking notes because everything's digital anyway

- Not turn in or copy others' assignments

Being remote actually makes it easier for me to tell

Recap: `if` statements

```
if (CONDITION) {
  // Code to execute
}
```

```
Example:
```

```
int apples = 5;
int oranges = 6;
if (if apples < oranges) {
  System.out.println("We have fewer apples!");
}
```
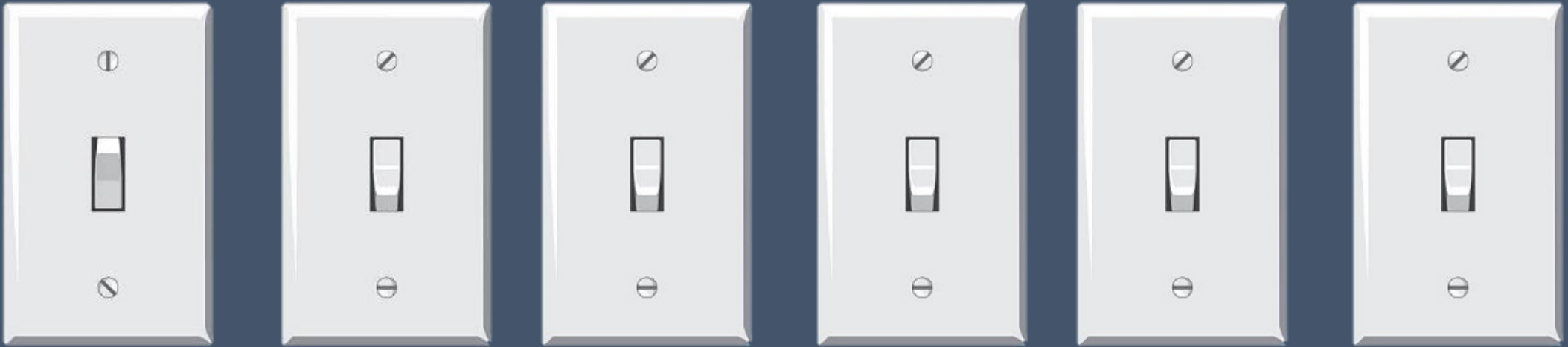
Recap: `if` statements

boolean expressions which use relational operators (like `>`) and/or logical operators (like `&&`).

If statements:

- Test boolean conditions for "truthiness" (e.g. expressions evaluating to true) and

- Execute statements based on the result of that evaluation

switch statements

```
Scanner input = new Scanner(System.in);
System.out.print("Enter either 1 or 2: ");

int num = input.nextInt();

switch(num) {
  case 1:
    System.out.println("You entered 1!");
    break;
  case 2:
    System.out.println("You entered 2!");
    break;
  default:
    System.out.println("You didn't follow directions!");
}
```

General form:

```
switch(VARIABLE) {
  case VALUE:
  // Thing to do
  break;
  default:
  // Thing to do if no match
}
```

switch vs. if statements

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

String fruit = input.nextLine();

if (fruit.equals("apple")
    || fruit.equals("pear")
    || fruit.equals("orange")) {
  System.out.println("Grows on a tree!");
} else if (fruit.equals("blueberry"
         || fruit.equals("raspberry")) {
  System.out.println("Grows on a shrub!");
} else {
  System.out.println("It's magic!");
}
```

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

String fruit = input.nextLine();

switch(fruit) {
  case "apple":
  case "pear":
  case "orange":
    System.out.println("Grows on a tree!");
    break;
  case "blueberry":
  case "raspberry":
    System.out.println("Grows on a shrub!");
    break;
  default:
    System.out.println("It's magic!");
    break;
}
```

switch vs. if statements

```
switch(fruit) {
   case "apple":
   case "pear":
   case "orange":
      System.out.println("Grows on a tree!");
      break;
   case "blueberry":
   case "raspberry":
      System.out.println("Grows on a shrub!");
      break;
   default:
      System.out.println("It's magic!");
      break;
}
```

Compares a single value's *equality* given a number of "cases."

Allows code to "break" (i.e. stop execution and return to the regular flow of control) if a give case matches.

Uses a "backstop" or default case in the event that nothing matches.

switch vs. if statements

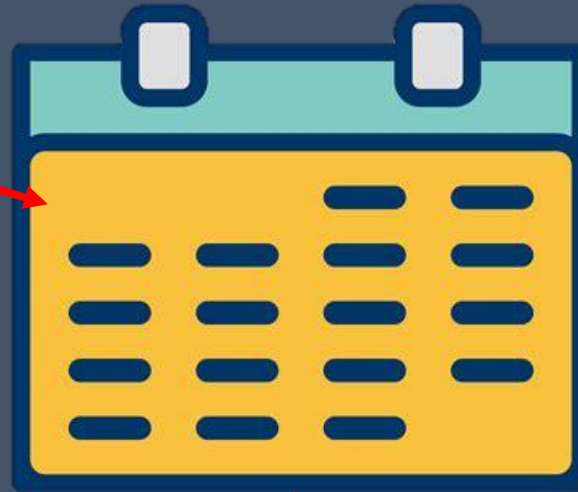| if statements	switch statements
Executes code based on the "truthiness" of a given expression	Executes code based on the value of a given variable
Can evaluate simple or complex expressions	Can only evaluate a single variable
Because boolean expression can evaluate nearly any data type, the if statement can evaluate a wider range of data	Can only evaluate **Strings**, **chars**, or **int** values
Very good for complicated situations where multiple conditions have to be true (e.g. "Prisoner's Dilemma")	Very good for single value scenarios (e.g. evaluating user input from a list of options)
Too complicated for evaluating simple equality	Too simple for evaluating complex conditions

Activity

cd to your copy of the class activities folder

perform a git pull download master

cd to the activity-10 folder to get started

Activity

12 months

4 seasons which change on their respective solstice/equinox

Max. 31 days (though February has 29 this year!)

Activity

Today, we're going to make a 2020 seasonal calendar to illustrate the difference between switch and if statements.

Here're our rules about the calendar:

- We take user input of a numeric month (1-12)

- We take user input of a day

- We take these inputs and determine:
  - The text name of the month (January, Februrary, et al.)

- Using these inputs, we assess if the season has changed:
  - Spring equinox: 20 March
  - Summer Solstice: 20 June
  - Autumn equinox: 22 September
  - Winter solstice: 21 December