CS Camp Review
Class, object, variable, type, method, if-then-else
Class: prototype defining the properties or attributes and actions of a particular type of object.
Class: prototype defining the properties or attributes and actions of a particular type of object.

Class example: human
Attribute exs: hair, color of hair (nouns)
Action exs: walk, read, write (verbs)
Class: prototype defining the properties or attributes and actions of a particular type of object.

Class example: human
Attribute exs: hair, color of hair (nouns)
Action exs: walk, read, write (verbs)

Java Class example: JButton
Attribute exs: size, location, text
Action exs: rollover, click
CS Camp Review
Class, object, variable, type, method, if-then-else

Object: entity that has unique existence among all objects, is of the type bearing the name of the Class from which it was created, and has attributes which may be instantiated at creation.
Object: entity that has unique existence among all objects, is of the type bearing the name of the Class from which it was created, and has attributes which may be instantiated at creation.

Object creation example: human
(x-rated material - deleted)
Object: entity that has unique existence among all objects, is of the type bearing the name of the Class from which it was created, and has attributes which may be instantiated at creation.

Object creation example: human
(x-rated material - deleted)

Java object creation example: JButton
new JButton("OK");
Variable: entity that is used to refer to an object or to hold an attribute value. There is no clear counterpart to a variable outside of engineering/science. Variables are allowed to refer only to objects of specified Classes.
Variable: entity that is used to refer to an object or to hold an attribute value. There is no clear counterpart to a variable outside of engineering/science. Variables are allowed to refer only to objects of specified Classes.

Java variable example:

```java
JButton button = new JButton("OK");
```
**Variable**: entity that is used to refer to an object or to hold an attribute value. There is no clear counterpart to a variable outside of engineering/science. Variables are allowed to refer only to objects of specified Classes.

Java variable example:

```java
JButton button = new JButton("OK");
button.setBackground(Color.blue);
```

Variables are needed to perform actions and change attributes after creation.
**Method:** Detailed description of some action that may be taken by an object of a class. Methods are the same for all objects of a class and are defined in the class prototype.
**CS Camp Review**

Class, object, variable, type, method, if-then-else

**Method:** detailed description of some action that may be taken by an object of a class. Methods are the same for all objects of a class and are defined in the class prototype.

Example for human:

**Walk:**

Repeat the following:
- lift left leg
- advance left leg by 50cm
- drop left leg
- lift right leg
- advance right leg by 50cm
- drop right leg
Method: detailed description of some action that may be taken by an object of a class. Methods are the same for all objects of a class and are defined in the class prototype.

Java example:

```java
public void changeToBlue (JButton b) {
    b.setBackground(Color.blue);
}
```
Method: detailed description of some action that may be taken by an object of a class. Methods are the same for all objects of a class and are defined in the class prototype.

Java example:

```java
public void changeToBlue (JButton b) {
    b.setBackground(Color.blue);
}
```

How to use a method within its class:

```java
JButton button = new JButton(“OK”);
changeToBlue(button);
```
if-then-else: first control statement in Java.

Syntax:
if (<condition>) {
    <statements>;
} else if (<condition>) {
    <statements>;
} else {
    <statements>;
}
if-then-else: first control statement in Java.

Example:

```java
if (evt.getSource() == b1) {
    b1.setText("X");
} else if (evt.getSource() == b2) {
    b2.setText("X");
} else {
    b3.setText("X");
}
```
Array: ordered collection of objects of the same type.

Declaration:

```java
JButton buttons[];
```
Array: ordered collection of objects of the same type.

Declaration:

JButton buttons[];

Creation of the array:

buttons = new JButton[9];
Array: ordered collection of objects of the same type.

Declaration:
JButton buttons[];

Creation of the array:
buttons = new JButton[9];

Creation of the JButtons:
buttons[0] = new JButton("First Button");
bbuttons[1] = new JButton("Second Button");
...
Array: ordered collection of objects of the same type.

Perform actions on the buttons:

```java
buttons[0].setText("OK");
bbuttons[1].setBackground(Color.blue);
```
for loop: repeat a collection of statements a given number of times.

Syntax:

for (<init> ; <condition> ; <increment>) {
    <statements>
}

CS Camp New Stuff
arrays, for loop, break
for loop: repeat a collection of statements a given number of times.

Example:

```java
for (int i=0 ; i < 9 ; i++) {
    button[i] = new JButton("X");
}
```
break: to break out of a loop early

Example:

```java
for (int i=0 ; i < 9 ; i++) {
    if (evt.getSource() == buttons[i] {  
        buttons[i].setText("X");
        break;
    }
}
```