SCSBC2 MIDTERMS

Java Conditional Statements

if

- used to specify a block of code to be executed, if a specified condition is true

Syntax:

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

else

- Used to specify a block of code to be executed, if the same condition is false

Syntax:

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

else if

- Used to specify a new condition to test, if the first condition is false

Syntax:

```
if (condition1) {
  // block of code to be executed if the condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition2 is true
} else {
  // block of code to be executed if the condition1 and
condition2 is false
}
```

switch

- Used to specify many alternative block of code to be executed
- Instead of writing many if...else statements, you can use the switch statement.

Syntax:

```
switch (expression) {
   case x:
      // code block
      break;
   case y:
      // code block
      break;
   default:
      // code block
}
```

- This is how it works:
  - The switch expression is evaluated once.
  - The value of the expression is compared with the values of each case.
  - If there is a match, the associated block of code is executed.
  - The break and default keywords are optional,

break keyword

- When Java reaches a break keyword, it breaks out of the switch block.
- This will stop the execution of more code and case testing inside the block.
- When a match is found, and the job is done, it's time for a break. There is no need for more testing.

default keyword

- The default keyword specifies some code to run if there is no case match:

Good luck, CS 1-2!