**switch-case**

**Example:**

**switch (var){**

**case 1:**

**//statements;**

**break;**

**case 2:**

**//statements;**

**break;**

**…**

**case n:**

**//statements;**

**break;**

**default:**

**//statements;**

**break;**

**}**

**Comparison:**

Switch uses the value stored in the passed variable to compare it to the values of each case. And if they are the same the following statements execute until they run into the break statement. If none of the cases match it goes to default case. If we miss a break while writing it continues to

**if else if else if ………**

**Example:**

**if (condition){**

**//Statements executes if condition //is true**

**}else if (condition2){**

**//Statements executes if condition2 //is true**

**}else if(conditionN){**

**//Statements executes if //conditionN is true**

**}else{**

**//Statements execute if all other //conditions are false**

**}**

**Comparison:**

If else uses conditions to check if the statements should be executed. If else is slower than switch, because if else has to go through all the conditions. Conditions can contain comparison operators and logical operators.

If else can take logical operators in as a condition as well as any expression or value that can evaluate to true. If else statement is harder to edit than a switch statement. When there is a lot of them. Else part is optional. That means if the condition inside the parentheses is false nothing will happen and the program will just continue to run normally.

compare to the other cases. Switch is faster than if else because when switch runs our compiler creates a jump table for switch so it skips the cases that are not correct and it is easier to troubleshoot than if else. You can also pass Boolean value as an argument and use condition instead of case values to create similar functionality to an if statement. This works because if the condition is true the true passed in as an argument will evaluate to true and the case will be executed.

**switch (true){**

**case (condition):**

**//statements;**

**break;**

**case (condition2):**

**//statements;**

**break;**

**….**

**case (condition n):**

**//statements;**

**break;**

**default:**

**//statements;**

**break;**

**}**

This is not the intended use of switch and it can leave someone reading your code wondering about what they are not seeing and also the compiler doesn’t use jump tables if we do this.