**What is the difference between an if statement and a switch statement?**

An *if-then-else* statement is preferable when we need to check ranges of values or multiple conditions.

A *switch* statement is better suited when testing a single variable against many single values. If you have to use relational or logical operators then you can’t use switch case.

**Describe the *if-then* and *if-then-else* statements. What types of expressions can be used as conditions?**

Both statements tell our program to execute the code inside of them only if a particular condition evaluates to *true*. However, the *if-then-else* statement provides a secondary path of execution in case the if clause evaluates to *false*.

**Describe the *switch* statement. What object types can be used in the *switch* clause?**

Switch allows the selection of several execution paths based on a variables’ value.

Each path is labelled with *case* or *default*, the *switch* statement evaluates each *case* expression for a match and executes all statements that follow the matching label until a *break* statement is found. If it can’t find a match, the *default* block will be executed instead

**What happens when we forget to put a *break* statement in a *case* clause of a *switch*?**

This means that it will continue the execution of all *case* labels until if finds a *break* statement, even though those labels don’t match the expression’s value.

**Can we use Switch statement with Strings?**

Prior to Java 7 we can use only int values and enum constants in Switch. Starting with Java 7 we can use strings in Switch statement. If we use strings in switch statement prior to Java 7 we will get compile time error “only int and enum constants are permitted”.