A - 7: Control structures to make decisions (selections)

Control structure	Description
Simple if construct	**if (** *boolean expression* **)** **{** One or more statements that will be executed if the boolean expression, given in parentheses above, is true. These statements will not be executed at all if the boolean expression is false (i.e. not true). **}**
if-else construct	**if (** *boolean expression* **)** **{** One or more statements that will be executed if the boolean expression, given in parentheses above, is true. **}** **else** **{** One or more statements that will be executed if the boolean expression, given in parentheses above, is false (i.e. not true). **}**
if-else if ... con-struct	**if (** *boolean expression 1* **)** **{** One or more statements that will be executed if and only if boolean expression 1 is true. **}** **else if (** *boolean expression 2* **)** **{** One or more statements that will be executed if and only if boolean expression 2 is true and boolean expression 1 is false. **}** **else** **{** One or more statements that will be executed if and only if neither boolean expression 1 nor boolean expression 2 is true. **}**
switch-case construct	**switch (** *arithmetic expression* **)** **{** **case** v_1**:** Statements which will be executed if the arithmetic expression has value v_1 **break ;** **case** v_2**:** Statements which will be executed if the arithmetic expression has value v_2 **break ;** **case** v_n**:** Statements to be executed when the arithmetic expression has value v_n **break ;** **default:** Statements which will be executed if none of the cases matched the value of the arithmetic expression **break ;** **}**

A - 8: Control structures to perform repetitions (iterations)

Control structure	Description
`while loop`	**`while (`** *boolean expression* **`)`** **`{`** One or more internal statements that will be repeatedly executed as long as the boolean expression, given in parentheses above, is true. **`}`**
`do-while loop`	**`do`** **`{`** One or more statements that will be first executed once, and then repeatedly executed as long as the boolean expression, given below in parentheses, is true. **`}`** **`while (`** *boolean expression* **`) ;`**
`for loop`	**`for (`** *assignment statement* **`;`** *boolean expression* **`;`** *increment or decrement statement* **`)`** **`{`** One or more internal statements that will be repeatedly executed as long as the boolean expression given above is true. When the boolean expression becomes false, the statements that follow this **`for`** loop will be executed. **`}`** An index variable may be declared in a **`for`** loop in the following way **`for ( int some_index = 0 ;`** **`...`** The scope of this kind of variable is within the internal statements of the loop.
`foreach loop`	**`foreach ( Type object_name in collection_name )`** **`{`** One or more statements that will be executed for each object in the collection. **`object_name`** refers to the object currently being processed, and the loop automatically processes all objects of the collection. The collection being processed can be a conventional array, an **`ArrayList`** array, or some other kind of collection that implements the **`IEnumerable`** interface. **`}`**