Core Java Part 2

Control Flow Statements(Selection Statements)

Trainer :Sanjay Kumar *Mobile:91-9818254421, 9910587200* *Email id:javaconsultant.sanjaykumar@gmail.com*

Objectives

- Decision making Statements
- ? As Conditional Operator
- if .. else
- switch .. case

Control Flow Statements

- All application development environments provide a decision making process called control flow statements that direct the application execution.

- Control flow enables a developer to create an application that can examine the existing conditions, and decide a suitable course of action.

Types of Control Flow Statements

- Decision-making (Selections)
  - ? As Conditional Statement
  - If - else statement
  - Switch - case statement

Control Structures

Work the same as in C / C++

If .. else and switch .. case

```
if(a > 3) {
    c = a;
}
else {
    c = 3;
}
```

```
C=(a > 3)?a:3;
```

```
switch(a) {
    case 1:
        c = 3;
    case 2:
        c = 3;
 case 4:
        c = a;
    default:
        c = 3;
}
```

If-else Statement

- The if-else statement tests the result of a condition, and performs appropriate actions based on the result.
- It can be used to route program execution through two different paths. \
- The syntax of if-else statement is::

```
if (condition)
{
        action1;
}
else
{
        action2;
}
```

Give this a Try…

What do you think is the output if a Number is 3?

```
if (aNumber >= 0){


 if (aNumber == 0)
   System.out.println("first string");
else
 System.out.println("second string");
 System.out.println("third string");
}
```

Switch – Case Statement

The switch – case statement can be used as an alternative for if-else-if statement.

- It is used in situations where an expression is evaluated multiple values.

- The use of the switch-case statement results in better performance.

- The syntax of switch-case is:

```
    switch (expression) {
 case 1:
    action1 statements;
    break;
 case 2':
    action2statements;
    break;
    ….
 case N ':
 actionN statements;
 break;
 default:
 default statements;
  }
```

Thank You