D4. Flow of Control & for Loop

switch statement

- A multiway *if* - *else* statement with many choices can be difficult to understand.
- A switch statement can be used in its place, when the choice is based on the value of an integer or character expression.
- It makes the code easy to understand.

A switch statement begins with the keyword switch followed by a controlling expression in parentheses:

```
switch (Controlling_Expression)
{
   case {\em Case_Label1:}
         {\em List_of_statements}
   case {\em Case_Label2:}
         {\em List_of_statements}
   . . .
}
```

- The body of the switch statement is enclosed in a pair of braces.

- In the following example, dayOfweek is the variable of the controlling expression.
- Within the braces we have a list of cases which correspond to the first (1), second(2) and on day of week.
- The switch statement is showing the classes I lecture on the first four days of the week.
- The last three days of the week, I have no classes.
- Since my schedule for Monday (day 1) and Wednesday(day 3) is identical, those two days (case 1: and case: 3) are written together.
- The action of each case ends with a break statement, the word break followed by a semicolon.
- When the execution reaches a break statement, the switch statement's execution ends.
- The break statement where a case statement ends.
- If there is no break statement separating one case from the next, both cases will be executed even if only one case is true.
- So please, separate cases with a break statement.

Eample

```
switch (dayOfWeek)
{
   case 1:
   case 3:
      System.out.println("CS212, CS220");
      break;
   case 2:
   case 4:
      System.out.println("CS313, CS220, CS111");
      break;
   default:
      System.out.println("No classes");
      break;
}
```

Random Class

- Java has two ways to generate Random numbers.
- (1) Using Random class to generate integer types.
- (2) Math.random() to generate a double number greater than equal to 0.0 and less than 1.0.

Random class

- To use Random class, we have to import java.util.Random.
- We need to create an object of the Random class to create a random number.

Examples

```
public class RandomDemo{
   public static void int main(String[] args)
   {
      Random rand = new Random();

      for(int i=0; i< 10; i++){
          System.out.print(rand.nextInt(30) + " ");
      }
      System.out.println();
}
```

- Java by default provides a seed. So we do not have to specifically seed it.
- The numbers generated are pseudo random numbers.
- The above example generates a random number from 0 to 29.
- If we need 30 as well, we have to change 30 to 31.

for Loop

- When we know the number of repitions we use a *for* loop.
- Java *for* loop is identical to C++.
- Although Java does not have a comma operator, the for statement in java has been defined so as to allow the comma operator.

```
for(int i=1, product = 1; i <= 10 : i++)
    product = product * i;
```

- This comma operator can only be used in a *for* loop.
- In the above example, i is the loop controlling variable. In general, it is better to have a body of a loop where all updates are made. It doesn't make sense to write both control and loop statements in the header line of a for loop. It can be both confusing, and can have errors.