Java Loop Control & Decision Making Statement

if...else Example

```
if (radius >= 0) {
  area = radius * radius * 3.14159;

     System.out.println("The area for the“   +
           “circle of radius " + radius + " is " + area);
}

else {
  System.out.println("Negative input");
}
```

Multiple Alternative if Statements

```
if (score >= 90.0)
  grade = 'A';
else
  if (score >= 80.0)
    grade = 'B';
  else
    if (score >= 70.0)
      grade = 'C';
    else
      if (score >= 60.0)
        grade = 'D';
      else
        grade = 'F';
```

Equivalent

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

Switch Statements

```
switch (status) {
  case 0:  compute taxes for single filers;
           break;
  case 1:  compute taxes for married file jointly;
           break;
  case 2:  compute taxes for married file separately;
           break;
  case 3:  compute taxes for head of household;
           break;
  default: System.out.println("Errors: invalid status");
}
```

Switch-case statement Example:

```
import java.util.Scanner;
public class HelloCSE {
public static void main(String[] args) {
   char grade;
   Scanner a = new Scanner(System.in);
   grade = a.next().charAt(0);
  switch (grade) {
   case 'A':
         System.out.println("Excellent!");
         break;
   case 'B':
        System.out.println(“Well done");
        break;
   case 'C':
         System.out.println(“Good");
         break;
   case 'D':
        System.out.println("You passed");
        break;
    case 'F':
         System.out.println("Better try again");
          break;
    default:
          System.out.println("Invalid grade");
          }
      System.out.println("Your grade is " + grade);
    }
}
```

Loop

- Java has very flexible three looping mechanisms. You can use one of the following three loops:
    - while Loop
    - do...while Loop
    - for Loop

while Loop

- A while loop is a control structure that allows you to repeat a task a certain number of times.

Syntax:

The syntax of a while loop is:

```
while(Boolean_expression)
 {
        //Statements
 }
```

while Loop Example:

```
public class Test{
public static void main(String args[]){
    int x =10;
    while( x <20){
        System.out.print("value of x : "+ x );
        x++;
        System.out.print("\n");
     }
  }
}
```

Output:

```
value of x :10
value of x :11
value of x :12
value of x :13
value of x :14
value of x :15
value of x :16
value of x :17
value of x :18
value of x :19
```

do...while Loop

The do...while Loop:

- A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

Syntax:

The syntax of a do...while loop is:

```
do
{
        //Statements
 } while(Boolean_expression);
```

do...while Loop Example:

```
public class Test{
public static void main(String args[]){
      int x =10;
     do{
            System.out.print("value of x : "+ x );
             x++;
             System.out.print("\n");
           }while( x <20);
     }
}
```

Output:

```
value of x :10
value of x :11
value of x :12
value of x :13
value of x :14
value of x :15
value of x :16
value of x :17
value of x :18
value of x :19
```

for Loop

- A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.
- A for loop is useful when you know how many times a task is to be repeated.

Syntax:

The syntax of a for loop is:

```
        for(initialization ; Boolean_expression ; update)
        {
                //Statements
        }
```

for Loop Example:

```
public class Test{
public static void main(String args[]){
    for(int x =10; x <20; x = x+1){
         System.out.print("value of x : "+ x );
         System.out.print("\n");
       }
   }
}
```

Output:

```
value of x :10
value of x :11
value of x :12
value of x :13
value of x :14
value of x :15
value of x :16
value of x :17
value of x :18
value of x :19
```

Enhanced for loop in Java:

Syntax:

The syntax of enhanced for loop is:

```
for(declaration : expression)
{
      //Statements
}
```

- **Declaration**: The newly declared block variable, which is of a type compatible with the elements of the array you are accessing. The variable will be available within the for block and its value would be the same as the current array element.
- **Expression**: This evaluates to the array you need to loop through. The expression can be an array variable or method call that returns an array.

Enhanced for loop Example:

```
public class Test{
public static void main(String args[]){
       int[] numbers ={10,20,30,40,50};
       for(int number : numbers ){
                System.out.print(number);
                System.out.print(", ");
         }
               System.out.print("\n");
     String[] names ={"James","Larry","Tom","Lacy"};
     for(String name : names ){
             System.out.print( name );
            System.out.print(", ");
      }
   }
}
```

```
Output:
10, 20, 30, 40, 50,
James, Larry, Tom, Lacy,
```