Chapter- 5

5.1 Fill in the blanks in each of the following statements:

a) Typically, **For** statements are used for counter-controlled repetition and **while** statements for sentinel-controlled repetition.
b) The do…while statement tests the loop-continuation condition **after** executing the loop's body; therefore, the body always executes at least once.
c) The **switch** statement selects among multiple actions based on the possible values of an integer variable or expression.
d) The **Continue** statement, when executed in a repetition statement, skips the remaining statements in the loop body and proceeds with the next iteration of the loop.
e) The **&& (conditional AND)** operator can be used to ensure that two conditions are *both* true before choosing a certain path of execution.
f) If the loop-continuation condition in a for header is initially **false** the program does not execute the for statement's body.
g) Methods that perform common tasks and do not require objects are called **static** methods.

5.2 State whether each of the following is *true* or *false*. If *false*, explain why.

a) The default case is required in the switch selection statement.
b) False. The default case is optional. If no default action is needed, then there's no need for a default case.
c) The break statement is required in the last case of a switch selection statement.
False. The break statement is used to exit the switch statement. The break statement is not required for the last case in a switch statement.

c) The expression ((x > y) && (a < b)) is true if eitherx > yis true ora < bis true.
False. Both of the relational expressions must be true for the entire expression to be true when using the && operator.

d) An expression containing the || operator is true if either or both of its operands are true.
True.

e) The comma (,) formatting flag in a format specifier (e.g., %,20.2f) indicates that a value should be output with a thousands separator.
True

f) To test for a range of values in a switch statement, use a hyphen (–) between the start and end values of the range in a case label.
False. The switch statement does not provide a mechanism for testing ranges of values, so every value that must be tested should be listed in a separate case label.

g) Listing cases consecutively with no statements between them enables the cases to perform the same set of statements.

True.

5.3 *(Write a Statement)* Write a Java statement or a set of Java statements to accomplish each of the following tasks:

a) Sum the odd integers between 1 and 99, using a for statement. Assume that the integer variables sum and count have been declared.

Ans:

```
sum = 0;
for ( count = 1; count <= 99; count += 2 )
sum += count;
```

b) Calculate the value of 2.5 raised to the power of 3, using the pow method.

Ans :

```
double result = Math.pow( 2.5, 3 );
```

c) Print the integers from 1 to 20, using a while loop and the counter variable i. Assume that the variable i has been declared, but not initialized. Print only five integers per line. [*Hint:* Use the calculation i % 5.When the value of this expression is 0, print a newline character; otherwise, print a tab character. Assume that this code is an application. Use the System.out.println() method to output the newline character, and use the System. out.print('\t') method to output the tab character.]

Ans:

```
i = 1;
while ( i <= 20 )
{
System.out.print( i );
if ( i % 5 == 0 )
System.out.println();
else
System.out.print( '\t' );
++i;
}
```

e) Repeat part (c), using a for statement

Ans :

```
for ( i = 1; i <= 20; i++ )
{
        System.out.print( i );

if ( i % 5 == 0 )
System.out.println();
```

```
else
System.out.print( '\t' );
        }.
```

5.4 (Find the Error) Find the error in each of the following code segments, and explain how to correct it:

```
a) i = 1;
while ( i <= 10 );
++i;
}
```

Ans :

Error: The semicolon after the while header causes an infinite loop, and there's a missing left brace.

Correction: Replace the semicolon by a {, or remove both the ; and the }.

```
b) for ( k = 0.1; k != 1.0; k += 0.1 )
System.out.println( k );
```

Ans :

Error: Using a floating-point number to control a for statement may not work, because floating-point numbers are represented only approximately by most computers.

Correction: Use an integer, and performthe proper calculation in order to get the values you desire:

```
for ( k = 1; k != 10; k++ )
System.out.println( (double) k / 10 );
```

```
c) switch ( n )
{
case 1:
System.out.println( "The number is 1" );
case 2:
System.out.println( "The number is 2" );
break;
default:
System.out.println( "The number is not 1 or 2" );
break;
}
```

Ans :

Error: The missing code is the break statement in the statements for the first case.

Correction: Add a break statement at the end of the statements for the first case. This omission is not necessarily an error if you want the statement of case 2: to execute every time the case 1: statement executes.

d) The following code should print the values 1 to 10:

```
n = 1;
while ( n < 10 )
        System.out.println( n++ );
```

Ans :

Error: An improper relational operator is used in the while's continuation condition.

Correction: Use <= rather than <, or change 10 to 11.