ASSIGNMENT 3

Snippet 1:

```
public class InfiniteForLoop {
public static void main(String[] args) {
for (int i = 0; i < 10; i--) {
System.out.println(i);
}
}
}
```

// Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?

Ans :-

The given Java program snippet has an infinite loop due to the incorrect update of the loop control variable in the for loop. To prevent the loop from running indefinitely, you need to adjust the update condition to increment i instead of decrementing it.

Correct code :-

```
public class InfiniteForLoop {
public static void main(String[] args) {
for (int i = 0; i < 10; i++) {
 System.out.println(i);
} } }
```

Snippet 2:

```
public class IncorrectWhileCondition {
public static void main(String[] args) {
int count = 5;
while (count = 0) {
System.out.println(count);
count--;
}
```

}

}

// Error to investigate: Why does the loop not execute as expected? What is the issue with the condition in the

`while` loop?

Ans :-

The while loop in the code does not execute as expected because the condition while (count = 0) uses an assignment operator (=) instead of a comparison operator (==). This causes a compilation error. We need to use 'while (count > 0)' to properly check the condition and decrement count until it reaches 0.

Correct code :-

```
public class IncorrectWhileCondition {

public static void main(String[] args) {

int count = 5;

while (count == 0) {

System.out.println(count);

count--;

}

}

}
```

Snippet 3:

```
public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num > 0);

}

}
```

// Error to investigate: Why does the loop only execute once? What is wrong with the loop condition in the `dowhile`

loop?

Ans:-

This loop runs infinite times because the condition always returns true. So we need to specify the condition that will terminate the loop under a certain condition.

Correct code:-

```
public class DoWhileIncorrectCondition {

public static void main(String[] args) {

int num = 0;

do {

System.out.println(num);

num++;

} while (num < 10);

}

}
```

Snippet 4:

```
public class OffByOneErrorForLoop {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

System.out.println(i);

}

// Expected: 10 iterations with numbers 1 to 10

// Actual: Prints numbers 1 to 10, but the task expected only 1 to 9

}

}
```

// Error to investigate: What is the issue with the loop boundaries? How should the loop be adjusted to meet the

expected output?

Ans :-

for (int i = 1; i <= 10; i++) runs the loop while i is less than or equal to 10 so it prints number from 1 to 10. If we want to print numbers from 1 to 9 then we need to modify the for loop as for (int i=1; i<10; i++)

Correct code :-

```
public class OffByOneErrorForLoop {

public static void main(String[] args) {

for (int i = 1; i < 10; i++) {

System.out.println(i);

}

}

}
```

Snippet 5:

```
public class WrongInitializationForLoop {

public static void main(String[] args) {

for (int i = 10; i >= 0; i++) {

System.out.println(i);

}

}

}
```

// Error to investigate: Why does this loop not print numbers in the expected order? What is the problem with the

initialization and update statements in the `for` loop?

Ans :-

The loop starts at 10 and checks if i >= 0 (which is true), but instead of decreasing i to count down to 0, the loop's update statement i++ increases i by 1. This results in an infinite loop because i will always be greater than or equal to 0 once it starts increasing. To print the numbers from 10 to 0 we need to modify the for loop as for (i=10; i>=0; i--).

Correct code :-

```
public class WrongInitializationForLoop {
public static void main(String[] args) {
for (int i = 10; i >= 0; i--) {
System.out.println(i);
}
}
}
```

Snippet 6:

```
public class MisplacedForLoopBody {
public static void main(String[] args) {
for (int i = 0; i < 5; i++)
System.out.println(i);
System.out.println("Done");
}
}
```

// Error to investigate: Why does "Done" print only once, outside the loop? How should the loop body be enclosed to include all statements within the loop?

Ans :-

This code prints ‘Done’ only once because it is not a part of for loop. If we don’t add curly braces after for loop , only the first statement following the for loop declaration will be considered the part of the loop. So if we want to execute multiple statements in a for loop we need to enclose them in curly braces.

Correct code :-

```
public class MisplacedForLoopBody {
public static void main(String[] args) {
for (int i = 0; i < 5; i++) {
System.out.println(i);
System.out.println("Done");
}
}
```

}

Snippet 7:

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

// Error to investigate: Why does this code produce a compilation error? What needs to be done to initialize the loop

variable properly?

Ans :-

This code produces a compilation error because we have not initialized the count variable before checking the condition in the while loop. So to avoid this error we need to initialize the count variable before executing the while loop.

Correct code :-

public class UninitializedWhileLoop {

public static void main(String[] args) {

int count=1;

while (count < 10) {

System.out.println(count);

count++;

}

}

}

Snippet 8:

```
public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num--;

} while (num > 0);

}

}
```

// Error to investigate: Why does this loop print unexpected numbers? What adjustments are needed to print the

numbers from 1 to 5?

Ans :-

The above code prints the numbers in unexpected order because it is incorrectly decrementing the num variable instead of incrementing it. The condition in do while loop must be num<=5 to print the numbers from 1 to 5.

Correct code :-

```
public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num++;

} while (num <= 5);

}

}
```

Snippet 9:

```
public class InfiniteForLoopUpdate {
```

public static void main(String[] args) {

for (int i = 0; i < 5; i += 2) {

System.out.println(i);

}

}

}

// Error to investigate: Why does the loop print unexpected results or run infinitely? How should the loop update

expression be corrected?

Snippet 10:

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

int num = 10;

while (num = 10) {

System.out.println(num);

num--;

}

}

}

// Error to investigate: Why does the loop execute indefinitely? What is wrong with the loop condition?

Ans :

The above code executes the loop indefinitely because the condition while(num=10) uses the assignment operator which assigns the value 10 to num and evaluates to 10. Instead of that we need the specify the condition that will terminate the loop under a certain condition.

Correct code :-

public class IncorrectWhileLoopControl {

public static void main(String[] args) {

```
int num = 10;
while (num > 0) {
System.out.println(num);
num--;
}
}
}
```

Snippet 11:

```
public class IncorrectLoopUpdate {
public static void main(String[] args) {
int i = 0;
while (i < 5) {
System.out.println(i);
i += 2; // Error: This may cause unexpected results in output
}
}
}
```

// Error to investigate: What will be the output of this loop? How should the loop variable be updated to achieve the

desired result?

Ans :-

Correct code :-

The output will be 0 2 4 because the i is incrementing by 2. To print the number from 0 to 4 we need to increment the variable i by 1.

Correct code :-

```
public class IncorrectLoopUpdate {
public static void main(String[] args) {
int i = 0;
while (i < 5) {
```

```
System.out.println(i);
i--;
}
}
}
```

Snippet 12:

```
public class LoopVariableScope {
public static void main(String[] args) {
for (int i = 0; i < 5; i++) {
int x = i * 2;
}
System.out.println(x); // Error: 'x' is not accessible here
}
}
```

// Error to investigate: Why does the variable 'x' cause a compilation error?

Ans :-

Variable x is declared inside the for loop so it is accessible only inside the for loop. If we want to access the value of x outside the for loop we need to define and initialize it outside the for loop

Correct code :-

```
public class LoopVariableScope {
public static void main(String[] args) {
int x=1;
for (int i = 0; i < 5; i++) {
x = i * 2;
}
System.out.println(x); // Error: 'x' is not accessible here
}
}
```

SECTION 2: Guess the Output

Snippet 1:

```
public class NestedLoopOutput {
public static void main(String[] args) {
for (int i = 1; i <= 3; i++) {
for (int j = 1; j <= 2; j++) {
System.out.print(i + " " + j + " ");
}
System.out.println();
}
}
}
```

// Guess the output of this nested loop.

Ans :-

Dry run steps :

i=1 j=1 output = 1 1

j=2 output = 1 1 1 2

i=2 j=1 output = 2 1

j=2 output = 2 1 2 2

i=3 j=1 output = 3 1

j=2 output = 3 1 3 2

Final output : 1 1 1 2

2 1 2 2

3 1 3 2

Snippet 2:

```
public class DecrementingLoop {
public static void main(String[] args) {
```

```
int total = 0;
for (int i = 5; i > 0; i--) {
total += i;
if (i == 3) continue;
total -= 1;
}
System.out.println(total);
}
}
```

// Guess the output of this loop.

Ans :-

Dry run steps :

i=5 total=5 total=4

i=4 total=8 total=7

i=3 total=10

i=2 total=12 total=11

i=1 total=12 total=11

Final output : 11

Snippet 3:

```
public class WhileLoopBreak {
public static void main(String[] args) {
int count = 0;
while (count < 5) {
System.out.print(count + " ");
count++;
if (count == 3) break;
}
System.out.println(count);
```

}

}

// Guess the output of this while loop.

Ans :-

Dry run steps :

1st iteration : output = 0

count = 1

2nd iteration : output = 0 1

count = 2

3rd iteration : output = 0 1 2

count = 3

Final output : 0 1 2 3

Snippet 4:

```
public class DoWhileLoop {
public static void main(String[] args) {
int i = 1;
do {
System.out.print(i + " ");
i++;
} while (i < 5);
System.out.println(i);
}
}
```

// Guess the output of this do-while loop.

Ans :-

Dry run steps :

1st iteration : output = 1

i = 2

2nd iteration : output = 1 2

count = 3

3rd iteration : output = 1 2 3

count = 4

4th iteration : output = 1 2 3 4

Count = 5

Final output : 1 2 3 4 5

Snippet 5:

```
public class ConditionalLoopOutput {
public static void main(String[] args) {
int num = 1;
for (int i = 1; i <= 4; i++) {
if (i % 2 == 0) {
num += i;
} else {
num -= i;
}
}
System.out.println(num);
}
}
```

// Guess the output of this loop.

Ans :-

Dry run steps :

i=1 num=0

i=2 num=2

i=3 num=-1

i=4 num=3

Final output : 3

Snippet 6:

```
public class IncrementDecrement {
public static void main(String[] args) {
int x = 5;
int y = ++x - x-- + --x + x++;
System.out.println(y);
}
}
```

// Guess the output of this code snippet.

Ans :-

++x = The value of x is incremented before it is used in an expression.

X++ = The value of x is incremented after it is used in an expression.

x-- = The value of x is decremented after it is used in an expression.

--x = The value of x is decremented before it is used in an expression.

Final output : 8

Snippet 7:

```
public class NestedIncrement {
public static void main(String[] args) {
int a = 10;
int b = 5;
int result = ++a * b-- - --a + b++;
System.out.println(result);
}
}
```

// Guess the output of this code snippet.

Ans :-

Final output : 49

Snippet 8:

```
public class LoopIncrement {
public static void main(String[] args) {
int count = 0;
for (int i = 0; i < 4; i++) {
count += i++ - ++i;
}
System.out.println(count);
}
}
```

// Guess the output of this code snippet.

Ans :-

Final output : -4

1. **Write a program to calculate the sum of the first 50 natural numbers.**

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int sum = 0;
      for (int i = 1; i <= 50; i++) {
         sum = sum + i;
      }
      System.out.println(sum);
   }
}
```

2. Write a program to compute the factorial of the number 10.

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int num = 10;
      int factorial = 1;
      for (int i = 1; i <= 10; i++) {
         factorial = factorial * i;
      }
      System.out.println(factorial);
   }
}
```

3. Write a program to print all multiples of 7 between 1 and 100.

Ans :-

```
public class Main {
   public static void main(String[] args) {
      for (int i = 7; i <= 100; i += 7) {
         System.out.println(i);
      }
   }
}
```

4. Write a program to reverse the digits of the number 1234. The output should be 4321.

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int number = 1234;
```

```
        int reversed = 0;
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number = number / 10;
        }
        System.out.println(reversed);
    }
}
```

5. Write a program to print the Fibonacci sequence up to the number 21.

Ans:-

```
public class Main {
    public static void main(String[] args) {
        int a = 0;
        int b = 1;
        System.out.print(a + " " + b);
        int next = a + b;
        while (next <= 21) {
            System.out.print(" " + next);
            a = b;
            b = next;
            next = a + b;
        }
        System.out.println();
    }
}
```

6. Write a program to find and print the first 5 prime numbers.

Ans:-

```
public class Main {
   public static void main(String[] args) {
      int count = 0;
      int number = 2;
      while (count < 5) {
         boolean isPrime = true;
         for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
               isPrime = false;
               break;
            }
         }
         if (isPrime) {
            System.out.print(number + " ");
            count++;
         }
         number++;
      }
   }
}
```

7. **Write a program to calculate the sum of the digits of the number 9876. The output should be 30 (9 + 8 + 7 + 6).**

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int number = 9876;
      int sum = 0;
```

```
        while (number != 0) {
            int digit = number % 10;
            sum += digit;
            number /= 10;
        }

        System.out.println(sum);
    }
}
```

8. Write a program to count down from 10 to 0, printing each number.

Ans:-

```
public class Main {
    public static void main(String[] args) {
        for (int i = 10; i >= 0; i--) {
            System.out.println(i);
        }
    }
}
```

9. Write a program to find and print the largest digit in the number 4825.

Ans:-

```
public class Main {
    public static void main(String[] args) {
        int number = 4825;
        int largestDigit = 0;
        while (number != 0) {
            int digit = number % 10;
            if (digit > largestDigit) {
```

```
            largestDigit = digit;
        }
        number /= 10;
    }
    System.out.println(largestDigit);
  }
}
```

10. Write a program to print all even numbers between 1 and 50.

Ans:-

```
public class Main {
  public static void main(String[] args) {
    for (int i = 1; i <= 50; i++) {
      if (i % 2 == 0) {
        System.out.println(i);
      }
    }
  }
}
```

11. Write a Java program to demonstrate the use of both pre-increment and post-decrement operators in a single expression

Ans :-

```
public class Main {
  public static void main(String[] args) {
    int x = 5;
    int y = 10;
    int result = ++x + y-- - x-- + ++y;
    System.out.println("Result: " + result);
    System.out.println("x: " + x);
```

```
        System.out.println("y: " + y);
    }
}
```

12. Write a program to draw the following pattern:

****

****

****

****

****

Ans:-

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                System.out.print("*");
            }
            System.out.println("");
        }
    }
}
```

13. Write a program to print the following pattern:

1

2*2

3*3*3

4*4*4*4

5*5*5*5*5

5*5*5*5*5

4*4*4*4

3*3*3

2*2

1

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int n = 5;
      for (int i = 1; i <= n; i++) {
         for (int j = 1; j <= i; j++) {
            if (j > 1)
               System.out.print("*");
            System.out.print(i);
         }
         System.out.println();
      }
      for (int i = n; i >= 1; i--) {
         for (int j = 1; j <= i; j++) {
            if (j > 1)
               System.out.print("*");
            System.out.print(i);
         }
         System.out.println();
      }
   }
}
```

14. Write a program to print the following pattern:

*

**

***

*****

****

****

Ans :-

```
public class Main {
   public static void main(String[] args) {
      for (int i = 1; i <= 6; i++) {
         for (int j = 1; j <= i; j++) {
            System.out.print("*");
         }
         System.out.println("");
      }
   }
}
```

15. Write a program to print the following pattern:

```
    *
   **
  ***
 ****
*****
```

Ans :-

```
public class Main {
   public static void main(String[] args) {
      int n = 5;
      for (int i = 1; i <= n; i++) {
         for (int j = 1; j <= (n - i); j++) {
```

```
            System.out.print(" ");
        }
        for (int k = 1; k <= i; k++) {
            System.out.print("* ");
        }
        System.out.println("");
    }
  }
}
```

16. Write a program to print the following pattern:

```
    *
   ***
  *****
*******
*********
```

Ans:-

```
public class Main {
   public static void main(String[] args) {
      int rows = 4;

      for (int i = 1; i <= rows; i++) {
         for (int j = rows; j > i; j--) {
            System.out.print(" ");
         }
         for (int k = 1; k <= (2 * i - 1); k++) {
            System.out.print("*");
         }
         System.out.println();
```

```
        }
    }
}
```

17. Write a program to print the following pattern:

```
*****
 ****
  ***
   **
    *
```

Ans:-

```
public class Star {

    public static void main(String[] args) {
        int rows = 5;
        for (int i = 0; i <= rows - 1; i++) {
            for (int j = 0; j <= i; j++) {
                System.out.print(" ");
            }
            for (int k = 0; k <= rows - 1 - i; k++) {
                System.out.print("*" + " ");
            }
            System.out.println();
        }

    }

}
```

18. Write a program to print the following pattern:

1

1*2

1*2*3

1*2*3*4

1*2*3*4*5

Ans:-

```
class Main {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                if (j > 1)
                    System.out.print("*");
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

20. Write a program to print the following pattern:

5

5*4

5*4*3

5*4*3*2

5*4*3*2*1

Ans :-

```
public class PatternPrinter {
```

```
    public static void main(String[] args) {
        int n = 5;

        for (int i = 0; i < n; i++) {
            for (int j = n; j > i; j--) {
                if (j < n) System.out.print("*");
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```

21. Write a program to print the following pattern:

1

1*3

1*3*5

1*3*5*7

1*3*5*7*9

Ans:-

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                if (j > 1) System.out.print("*");
                System.out.print(2 * j - 1);
            }
```

```
            System.out.println();
        }
    }
}
```

23. Write a program to print the following pattern:

11111

22222

33333

44444

55555

Ans :-

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

24. Write a program to print the following pattern:

1

22

333

4444

55555

Ans:-

```
public class Main {
   public static void main(String[] args) {
      int rows = 5;
      for (int i = 1; i <= rows; i++) {
         for (int j = 1; j <= i; j++) {
            System.out.print(i);
         }
         System.out.println();
      }
   }
}
```

25. Write a program to print the following pattern:

1

12

123

1234

12345

Ans:-

```
public class Main {
   public static void main(String[] args) {
      int rows = 5;
      for (int i = 1; i <= rows; i++) {
         for (int j = 1; j <= i; j++) {
            System.out.print(j);
         }
         System.out.println();
      }
```

```
    }
}
```

26. Write a program to print the following pattern:

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

Ans :-

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        int number = 1;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(number + " ");
                number++;
            }
            System.out.println();
        }
    }
}
```