CDAC MUMBAI

Lab Assignment

SECTION 1: Error-Driven Learning Assignment: Loop Errors

Instructions:

Analyze each code snippet for errors or unexpected behavior. For each snippet, determine:

1. Why does the error or unexpected behavior occur?
2. How can the code be corrected to achieve the intended behavior?

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:
1)In the iteration part (i) should be incremented.
2)Corrected code:

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

.

CDAC Mumbai

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:

1)Inside while condition we have assigned count value to o rather than giving condition like (!=) or(==);

2)Corrected 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 `do-while` loop?
```

Ans:

1**)in Do While Loop do block executes once even if condition is false.**

2)in do body after incremention of 0 to 1 condition becomes false. So it only printed once.

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:

1)inside loop condition we have given (i<=10) .so it will also print 10.

2)if we have to print only 1 to 9. Loop condition should be (i<=9);

3)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:

1)In iteration part the I should be decremented.

2) 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);
```

CDAC Mumbai
CDAC Mumbai

```
            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:

1)because second S.O.P Statement is inside the loop.

2)for enclosing we have to make second s.o.p statement inside curly bracket

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:

1)because code is not initialized.

2)for that we have to initialize i=0;

correct code:

```
public class UninitializedWhileLoop {
    public static void main(String[] args) {
        int count;
        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:

1)int num should be initialized as 5.if we initialized at 1 it will print only once.

correct code:

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

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?

Ans:

1)The loop will run.but only print 0,2 and 4.because in loop iteration condition we incresedit by 2.

2)updated loop.

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

CDAC Mumbai

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:

1) **We have assigned num to 0 not used (==) or(>=) Operator.**
2) **For that we have to use(num>0):**
3) **Correct code:**

```
public class
IncorrectWhileLoopControl {
public static void main(String[]
args) {
int num = 10;
while (num =
10) {
System.out.println(nu
m); 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:

1)we are incrementing with 2 in while body.still it is giving output of 0 2 4.

2)for desired output we have to increment it by 1 .i.e i++.

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? How does scope

Ans:

1)X variable is initialized inside the loop so it will create error in s.o.p statement.

2)correct code:

```
public class LoopVariableScope {
    public static void main(String[] args) {
    int x;
    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

CDAC Mumbai
CDAC Mumbai

Instructions:

1. **Perform a Dry Run:** Carefully trace the execution of each code snippet manually to determine the output.
2. **Write Down Your Observations:** Document each step of your dry run, including the values of variables at each stage of execution.
3. **Guess the Output:** Based on your dry run, provide the expected output of the code.
4. **Submit Your Assignment:** Provide your dry run steps along with the guessed output for each code snippet.

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:

i	**j**	**i<=3**	**j<=2**	**output**
1	**1**	**T**	**T**	**“ 1 2”**
1	**2**	**T**	**T**	**“ 1 1 1 2”**
1	**3**	**T**	**F**	**“ 1 1 1 2” \n inner loop gets out of condition.**
2	**1**	**T**	**T**	**“ 1 1 1 2 \n 2 1”**
2	**2**	**T**	**T**	**“ 1 1 1 2 \n 2 1 2 2”**
2	**3**	**T**	**F**	**“ 1 1 1 2 \n 2 1 2 2”\n inner loop gets out of condition.**
3	**1**	**T**	**T**	**“ 1 1 1 2 \n 2 1 2 2 \n 3 1”**
3	**2**	**T**	**T**	**“ 1 1 1 2 \n 2 1 2 2 \n 3 1 3 2”**

Output will be:

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:

i	**i>0**	**Total+=**	**cndtn(i==3)**	**total -=1**	**total**
5	**T**	**0+5**	**F**	**5-1**	**4**
4	**T**	**4+4**	**F**	**8-1**	**7**
3	**T**	**7+3**	**T**	**F**	**10**
2	**T**	**10+2**	**F**	**12-1**	**11**
1	**T**	**11+1**	**F**	**12-1**	**12**

Output is: 11

CDAC Mumbai

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:

count	cndtn(count<5)	Inner S.O.P and Output	count++	countouterloop	count==3	break
0	T	" 0 "	1	F	F	
1	T	" 0 1"	2	F	F	
2	T	" 0 1 2"	3	T	T	
3	F	"0 1 2 \n 3 "				

Expected 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:

i(count)	output	i++	i(count)	condition(i<5)
1	" 1"	2	2	T
2	" 1 2"	3	3	T
3	" 1 2 3"	4	4	T
4	" 1 2 3 4"	5	5	F
5	" 1 2 3 4 \n 5"	-	5	-

Expected 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:

i	**cndtn(i%2==0)**	**innerloopoperation**	**num(before)**	**num(before)**
1	**F**	**else num-=**	**1**	**0**
2	**T**	**if num+=**	**0**	**2**
3	**F**	**else num-=**	**2**	**-1**
4	**4**	**if num+=**	**-1**	**3**

Expected Output= 3

CDAC Mumbai

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:

```
int y = ++x - x-- + --x + x++;
            6  -   6    +   4 +    4   = 8
```

Expected 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:

```
 int result = ++a * b---------- a + b++;
                    11  * 5    -        10  +   5
```

Expected output: 45.

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:

In iteration 1:
i=0;
i++ = 0;
++I = 2;
Count=0-2=- 2;
In iteration 2:
i=3;
i=5;
count=3-5=-2;
ans=-2+-2=-4.

Expected output=4;

SECTION 3: Lamborghini Exercise:

Instructions:

1. **Complete Each Program:** Write a Java program for each of the tasks listed below.
2. **Test Your Code:** Make sure your code runs correctly and produces the expected output.
3. **Submit Your Solutions:** Provide the complete code for each task along with sample output.

Tasks:

1. Write a program to calculate the sum of the first 50 natural numbers.
2. Write a program to compute the factorial of the number 10.
3. Write a program to print all multiples of 7 between 1 and 100.
4. Write a program to reverse the digits of the number 1234. The output should be 4321.
5. Write a program to print the Fibonacci sequence up to the number 21.
6. Write a program to find and print the first 5 prime numbers.
7. Write a program to calculate the sum o321f the digits of the number 9876. The output should be 30 (9 + 8 + 7 + 6).
8. Write a program to count down from 10 to 0, printing each number.
9. Write a program to find and print the largest digit in the number 4825.
10. Write a program to print all even numbers between 1 and 50.
11. Write a Java program to demonstrate the use of both pre-increment and post-decrement operators in a single expression
12. Write a program to draw the following pattern:

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

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
```

CDAC Mumbai

```
1
```

14. Write a program to print the following pattern:

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

15. Write a program to print the following pattern:

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

16. Write a program to print the following pattern:

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

17. Write a program to print the following pattern:

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

18. Write a program to print the following pattern:

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

19. Write a program to print the following pattern:

```
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
```

CDAC Mumbai

20. Write a program to print the following pattern:

```
5
5*4
5*4*3
5*4*3*2
5*4*3*2*1
```

21. Write a program to print the following pattern:

```
1
1*3
1*3*5
1*3*5*7
1*3*5*7*9
```

22. Write a program to print the following pattern:

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

23. Write a program to print the following pattern:

```
11111
22222
33333
44444
55555
```

24. Write a program to print the following pattern:

```
1
22
333
4444
55555
```

25. Write a program to print the following pattern:

```
1
12
123
1234
12345
```

26. Write a program to print the following pattern:

CDAC Mumbai

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

CDAC Mumbai