Name :Shivam Dolase_KH

Assignment 3 Solutions:

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?
```

Answer:
Loop runs infinitely because the value of iterator i keeps on decreasing due to decrementing operator.
Change it to incrementing operator.
Solution:

```
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?
```

Answer:
The condition in while loop was wrong

```
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?

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?

Answer :
The equal to sign in for loop condition i<=10 causes the 10 iterations and by removing and makink it i<10 we get 9 iteration i.e.1 to 9

```
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
    }
}
```

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?
```

Answer:
Here in for loop instead of incrementing operator for i i.e. i++ , we should use decrementing operator i.e. i++

```
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?
```

Answer:
“Done” print statement not included in the for loop thus printedonly once, we will have to apply parenthesis {} to include it inside.

```
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?

Answer:

The count dose not have a initial integer value allocated to it , as it is not properly defined it will produce a compilation error. Thus here by allocating int count=0 we resolve the error.

```
public class UninitializedWhileLoop {
    public static void main(String[] args) {
        int count=0;
        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?

Answer:

Here while condition must be changed from num>0 to num<=5, and num-- in the loop must be changed to num++.Thus the required output sequence is achived.

```
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?

Answer:

No error in it . it prints 0,2,4 and the terminates the output is correct according to the logic.

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?

Answer:

```
public static void main(String[] args) {
        int num = 1;
        while (num = 10) {
            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?

Answer:

```
public class IncorrectLoopUpdate {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            ++i;// now it will print 0 – 4 in expected order.
            //i += 2; // Error: This may cause unexpected results in output
        }
    }
}
```

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
```

Answer:

The variable x is the local variable of the for loop thus can not access it outside the for loop. Thus it give compilation error because program is trying to access it outside the scope.

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

SECTION 2: Guess the Output

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.
```

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.
```

Output:11

0+5=5-1=4+4=8-1=7+3=10+2=12-1=11+1=12-1=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.
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.
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.
Output:3
0+2=2-3=-1+4=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.
```

Output:8
6-6+4+4=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.
```

Output:49
11*5 – 10+4=55-10+4=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.
```

Output: -4

0-2=-2 i=2 ++ i=3
-2+3-5=-4 i=5 loop terminates.

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.

Solution:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){

            int n=50;
             int  sum=(n*(n+1))/2;
             System.out.println(sum);
      }
}
```

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

Solution:

```
import java.util.Scanner;
public class Demo{
    public static void main(String args[]){

            int n=10;
             int ans=1;
             for(int i=1;i<=n;i++){
                ans*=i;
             }

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

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

Solution:

```
import java.util.Scanner;
public class Demo1{
      public static void main(String args[]){

              int n=100;
              int i=7;
              while(i<=n){
                 System.out.print(i+" ");
                 i+=7;
              }


      }
}
OR
import java.util.Scanner;
public class Demo1{
      public static void main(String args[]){

              int n=100;
              int x=7;


              for(int i=x;i<=n;i++){
                 if(i%x==0){
                 System.out.println(i);
                 }
              }

      }
}
```

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

Solution:

```
import java.util.Scanner;
public class Demo1{
      public static void main(String args[]){

              int n;
              System.out.println("Enter a number");
              Scanner scanner = new Scanner(System.in);
              n = scanner.nextInt();


              while(n>0){
                 int  a=n%10;
                 n=n/10;
                 System.out.print(a);
              }
      }
}
```

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

Solution:

```
    import java.util.Scanner;
    public class Demo1{
      public static void main(String args[]){

          int n;
          System.out.println("Enter a number");
          Scanner scanner = new Scanner(System.in);
          n = scanner.nextInt();
          int a=0,b=1;

          while(a!=n){
            int temp=b;
            b=b+a;
            a=temp;
            System.out.print(a+" ");
          }


      }
    }
```

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

Solution:

```
    import java.util.Scanner;
    public class Demo1{
      public static void main(String args[]){

          int n=100;
          int count=0;
          boolean flag;

          for(int i=2;i<=n;i++){

            flag=true;

            for(int j=2;j<i;j++){
              if(i%j==0){
                flag=false;
                break;
              }
            }
            if(flag){
              System.out.print(i);
              count++;
            }
```

```
                    if(count==5){
                      break;
                    }
                  }

            }
        }
```

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

Code:

```
        import java.util.Scanner;
        public class A3_7{
            public static void main(String args[]){

                  int n;
                  System.out.println("Enter a number");
                  Scanner scanner = new Scanner(System.in);
                  n = scanner.nextInt();

                  int sum=0;
                  while(n>0){
                    int  a=n%10;
                    n=n/10;
                    sum+=a;
                  }
                 System.out.println(sum);
           }
        }
```

Output:

```
Enter a number
9876
30
```

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

```
        public class Demo1{
            public static void main(String args[]){
                  int n=10;

                  for(int i=10;i>=0;i--){
                    System.out.print(i+" ");
                  }

            }
}
```

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

Code:

```
        import java.util.Scanner;
        public class Demo1{
            public static void main(String args[]){
                int n;
                System.out.println("Enter a number");
                Scanner scanner = new Scanner(System.in);
                n = scanner.nextInt();

                int maxN=0;
                while(n>0){
                   int  a=n%10;
                   n=n/10;
                   maxN=Math.max(maxN,a);
                }
               System.out.println(maxN);
         }
     }
```

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

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

                int n=50;
                for(int i=1;i<=50;i++){
                   if(i%2==0){
                      System.out.print(i+" ");
                   }
                }

         }
}
```

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

Code:

```
     import java.util.Scanner;
     public class Demo1{
          public static void main(String args[]){
                int n=100;

                System.out.print((n++)+" "+(++n));

          }
     }
```

Output:

```
100 102
```

12. Write a program to draw the following pattern:

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

Solution:

```
    public class Demo1{
        public static void main(String args[]){
            int n=5;

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

              for(int j=1;j<=n;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
```

Solution:

```
    public class Demo1{
        public static void main(String args[]){
            int n=5;

            for(int i=1;i<=n;i++){
              for(int j=1;j<=i+(i-1);j++){
                if(j%2==0){
                  System.out.print("*");

                }
                else{
                  System.out.print(i);
                }
              }
        System.out.println();
       }
       for(int i=n;i>=1;i--){
        for(int j=1;j<=i+(i-1);j++){
         if(j%2==0){
          System.out.print("*");

         }
         else{
          System.out.print(i);
         }
        }
        System.out.println();
       }

     }
}
```

14.Write a program to print the following pattern:

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

Solution:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){

        int n;
         System.out.println("Enter a number");
         Scanner scanner = new Scanner(System.in);
         n = scanner.nextInt();

         for(int i=1;i<=n;i++){
             if(i%2==0&&i>2){
                 continue;
             }
             for(int j=1;j<=i;j++){
                 System.out.print("*");
              }
              System.out.println();
       }

   }
}
```

15. Write a program to print the following pattern:

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

Solution:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){
         int n=5;

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

            for(int j=1;j<=n;j++){
               if(j<=n-i){
                  System.out.print(" ");
               }
               else{
                  System.out.print("* ");
               }
            }
            System.out.println();
         }

    }
}
```

16. Write a program to print the following pattern:

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

Solution:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){
       int n=9;

       for (int i=1;i<=n;i++){
          if(i%2==0){
             continue;
          }
          for(int j=1;j<=((n-i)/2);j++){
             System.out.print(" ");
          }
          for(int j=1;j<=i;j++){
             System.out.print("*");
          }
```

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


    }
}
```

17. Write a program to print the following pattern:

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

Solution:

```
public class Demo1{
    public static void main(String args[]){
        int n=5;

        for(int i=n;i>=1;i--){

            for(int j=1;j<=n;j++){
                if(j<=n-i){
                    System.out.print(" ");
                }
                else{
                    System.out.print("* ");
                }
            }
            System.out.println();
        }

    }
}
```

18. Write a program to print the following pattern:

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

Solutions:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){
        int n=9;

        for (int i=1;i<=n;i++){
            if(i%2==0){
                continue;
            }
            for(int j=1;j<=((n-i)/2);j++){
                System.out.print(" ");
            }
            for(int j=1;j<=i;j++){
                System.out.print("*");
            }
            System.out.println();
        }
        for (int i=n-1;i>=1;i--){
            if(i%2==0){
                continue;
            }
            for(int j=1;j<=((n-i)/2);j++){
                System.out.print(" ");
            }
            for(int j=1;j<=i;j++){
                System.out.print("*");
            }
            System.out.println();
        }

    }
}
```

19. Write a program to print the following pattern:

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

Solution:

```
public class Demo1{
    public static void main(String args[]){
        int n=5;

        for(int i=1;i<=n;i++){
        int count=1;
            for(int j=1;j<=i+(i-1);j++){
                if(j%2==0){
                    System.out.print("*");

                }
                else{
                    System.out.print(count);
                    ++count;
                }
            }
            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
```

Solution:

```
public class Demo1{
    public static void main(String args[]){
        int n=5;

        for(int i=1;i<=n;i++){
        int count=n;
          for(int j=1;j<=i+(i-1);j++){
            if(j%2==0){
              System.out.print("*");

            }
            else{
              System.out.print(count);
              --count;
            }
          }
          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
```

Solution:

```
import java.util.Scanner;
public class Demo1{
    public static void main(String args[]){
        int n=5;

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

           for(int j=1;j<=i+(i-1);j++){
              if(j%2==0){
                 System.out.print("*");

              }
              else{
                 System.out.print(j);

              }
           }
           System.out.println();
        }



    }
}
```

22. Write a program to print the following pattern:

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

23. Write a program to print the following pattern:

```
11111
22222
33333
44444
55555
```

Solutions:

```
        public class Demo1{
              public static void main(String args[]){
                      int n=5;

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

                         for(int j=1;j<=n;j++){
                                System.out.print(i+ " ");
```

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

         }
}
```

24. Write a program to print the following pattern:

```
1
22
333
4444
55555
```

Solutions:

```
      public class Demo1{
         public static void main(String args[]){
             int n=5;

             for(int i=1;i<=n;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
```

Solutions:

```
      public class Demo1{
         public static void main(String args[]){
             int n=5;

             for(int i=1;i<=n;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
```

Solutions:

```
            public class Demo1{
                  public static void main(String args[]){
                          int n=5;
                          int count=1;
                          for(int i=1;i<=n;i++){

                              for(int j=1;j<=i;j++){
                                      System.out.print(count+ " ");
                                      ++count;
                              }
                              System.out.println();
                          }

                  }
}
```