**Day 3: Assignments**

**Section I : java code snippets:-**

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: Loop runs indefinitely because of i—to fix this replace with 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: Error , because condition inside while loop is wrong , by changing condition to count!=0 will fix this.

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: No, the loop runs infinitely, because terminating condition is not meeting.

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: yes, it prints from 1 to 10 perfectly.

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: This will run infinitely., condition should be ‘i<=10’.

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?

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?

// error: variable count might not have been initialized.

//count should be initialized with 0. in java local variables must be explicitly initialized before they are accessed or used in any operation.

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?

//num— should be num++ and at while loop 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?

// i is getting incremented by 2 in every iteration.

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?

// in while loop condition assignment operator is present , it should be ‘num != 0’.

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?

// the above code will print : 0 2 4

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: the x is local variable , it is not accessible outside the scope of the for loop.

**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.

i j

1 1

1 2

2 1

2 2

3 1

3 2

o/p: 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.

i total

5 4

4 7

3 10

2 11

1 11

0 loop ends..

o/p: total: 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.

o/p : 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.

//o/p: 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.

//o/p: 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.

//Y = 6 – 6 + 4 + 4;

//o/p: 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.

//result = 11 \* 5 – 10 + 4;

//o/p : 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.

//o/p: -4

**Section 3: Lamborgini Exercise**

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

Code:

SumOfNaturalNumbers {

public static void main(String args[]){

int sum = 0;

for(int i = 1; i<=50; i++){

sum += i;

}

System.out.println("Sum of first 50 natural numbers is: " + sum);

}

}

o/p: Sum of first 50 natural numbers is: 1275

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

Code:

class FactoriaOf\_10 {

public static void main(String args[]){

long fact = 1;

for(int i = 10; i >= 1; i--){

fact \*= i;

}

System.out.println("Factorial of 10 is: " + fact);

}

}

o/p: Factorial of 10 is: 3628800

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

Code:

class MultiplesOf\_7 {

public static void main(String args[]){

int num = 7;

for(int i = 1; i <= 100; i++){

if(i % num == 0){

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

}

}

}

}

o/p: 7 14 21 28 35 42 49 56 63 70 77 84 91 98

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

Code:

class ReverseDigit {

public static void main(String args[]){

int num = 1234;

int rev = 0;

while(num != 0){

int digit = num % 10;

rev = rev \* 10 + digit;

num /= 10;

}

System.out.println("Reversed number is: " + rev);

}

}

o/p: Reversed number is: 4321

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

Code:

class FibonacciSequence {

public static void main(String args[]){

int num1 = 0, num2 = 1;

System.out.print("Fibonacci sequence up to 21 : ");

while(num1 <= 21){

System.out.print(num1 + " ");

int nextNum = num1 + num2;

num1 = num2;

num2 = nextNum;

}

}

}

o/p: Fibonacci sequence up to 21 : 0 1 1 2 3 5 8 13 21

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

Code:

class FindPrime {

public static void main(String[] args) {

int count = 0;

int num = 2;

System.out.print("First 5 prime numbers are: ");

while(count < 5){

if(isPrime(num)){

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

count++;

}

num++;

}

}

public static boolean isPrime(int num){

if(num <=1)

return false;

for(int i = 2; i<= num / 2; i++){

if(num % i == 0){

return false;

}

}

return true;

}

}

o/p : First 5 prime numbers are: 2 3 5 7 11

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

Code:

class SumOfNumber {

public static void main(String args[]){

int num = 9876;

int sum = 0;

while(num != 0){

int rem = num % 10;

sum += rem;

num /= 10;

}

System.out.println("Sum of digit of a number is: " + sum);

}

}

o/p: Sum of digit of a number is: 30

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

Code:

class CountDown {

public static void main(String args[]){

int n = 10;

while(n != 0){

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

n--;

}

}

}

o/p: 10 9 8 7 6 5 4 3 2 1

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

Code:

Class LargestDigit {

public static void main(String args[]){

int num = 4825;

int largest = 0;

while(num != 0){

int rem = num % 10;

if(rem > largest){

largest = rem;

}

num = num / 10;

}

System.out.println("largest digit of the number: " + largest);

}

}

o/p: largest digit of the number: 8

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

Code:

Class PrimeNumbers {

public static void main(String args[]){

System.out.print("All prime numbers form 1 to 50 : ");

for(int i = 1; i<=50; i++){

if(i % 2 == 0){

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

}

}

}

}

o/p: All prime numbers from 1 to 50 : 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

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

Code:

public class IncrementDecrement {

public static void main(String args[]) {

int a = 10;

int b = --a + a-- + ++a + a--;

System.out.println("Value of a is: " + a);

System.out.println("Value of b is: " + b);

}

}

o/p: Final value of a: 8

Value of b: 36

1. Pattern 1:

Code:

class Pattern1 {

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();

}

}

}

o/p: \*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

1. Pattern2:

Code:

class Pattern2 {

public static void main(String[] args) {

for(int i = 1; i<=5; i++){

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

if(j>1){

System.out.print("\*");

}

System.out.print(i);

}

System.out.println();

}

for(int i = 5; i>=1; i--){

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

if(j>1){

System.out.print("\*");

}

System.out.print(i-1);

}

System.out.println();

}

}

}

o/p: 1

2\*2

3\*3\*3

4\*4\*4\*4

5\*5\*5\*5\*5

4\*4\*4\*4

3\*3\*3

2\*2

1

1. Pattern3:

Code:

class Pattern3 {

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();

}

}

}

o/p: \*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

1. Pattern4:

Code:

class Pattern1 {

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();

}

}

o/p: \*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

1. Patter5:

Code:

public class CenterAlignedPyramidPattern {

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 <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

o/p:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

1. Pattern6:

Code:

public class Pattern6 {

public static void main(String[] args) {

int n = 5;

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

}

}

}

o/p:

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

1. Pattern7:

Code:

public class Pattern7{

public static void main(String[] args) {

int n = 4;

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

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

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

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

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

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

o/p:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

1. Pattern8:

Code:

public class NumberAsteriskPattern {

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(j);

}

System.out.println();

}

}

}

o/p:

1

1\*2

1\*2\*3

1\*2\*3\*4

1\*2\*3\*4\*5

1. Pattern9:

Code:

public class Pattern9 {

public static void main(String[] args) {

int n = 5;

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

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

if (j < n) {

System.out.print("\*");

}

System.out.print(j);

}

System.out.println();

}

}

}

o/p:

5

5\*4

5\*4\*3

5\*4\*3\*2

5\*4\*3\*2\*1

1. Pattern10:

Code:

public class OddNumberAsteriskPattern {

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(2 \* j - 1);

}

System.out.println();

}

}

}

o/p:

1

1\*3

1\*3\*5

1\*3\*5\*7

1\*3\*5\*7\*9

1. Pattern11:

Code:

public class HourglassPattern {

public static void main(String[] args) {

int n = 5;

//for Upper Pyramid

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

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

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

// for Lower pyramid

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

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

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

o/p:

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

1. Pattern12:

Code:

public class RepeatedNumberPattern {

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();

}

}

}

o/p:

11111

22222

33333

44444

55555

1. Pattern13:

Code:

public class RepeatedNumberPattern {

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();

}

}

}

o/p:

1

22

333

4444

55555

1. Pattern14:

Code:

public class RepeatedNumberPattern {

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();

}

}

}

o/p:

1

12

123

1234

12345

1. Pattern15:

Code:

public class RepeatedNumberPattern {

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();

}

}

}

o/p:

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15