Assignment 3

Section 1

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?

Solution--— The above code runs infinitely because the variable ‘i’ in for loop starts with a initial value 0, and the condition to be checked is i<10 i.e if i is lesser than the 10 after which the value of i is further decreased by doing i- - making the previous condition to be always true. This makes the loop run infinitely .The loop control should be set to i++ rather than i- - to terminate it after finite execution.

Corrected 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?

Solution--—The condition in while() loop in java should always return a Boolean value-true or false. In the above code the while() has a assignment expression rather than a condition to be checked which is why the code generates an error.

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 `dowhile` loop?

Solution--—The above code works perfectly fine.There is nothing wrong with the loop condition .Had it been –

do {

System.out.println(num);

num--;

} while (num > 0);

then it would have run only one time.

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?

Solution--—The result of this code is expected as it prints values from 1 to 10.No mismatch in expected and actual result. If the expected result was 1 to 9 condition in for() should change to for (int i = 1; i < 10; 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?

Solution--—This loops starts printing numbers from 10 and runs infinitely .Assuming that the loop should print values 1 to 10.Conditions of for() should change to –

public class WrongInitializationForLoop {

public static void main(String[] args) {

for (int i = 1; i <=10; 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?

Solution--—For a loop in java , if no curly brackets are used in this definition then only the next line after the loop is considered its part.In the above code no curly brackets are used for the for() loop and hence only the statement after that is considered its part.The statement to print “Done” falls outside the for() loop scope therefore it is only printed once .For it be printed multiple times it should be included in the for() loop by using curly brackets.

Corrected 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?

Solution--—the above code produces error because before the variable “count” is used to check condition in while() it is not assigned any value i.e it is not been initialized .

To make it work “count” should be initialized.

Corrected Code—

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?

Solution--the above code prints only one value i.e 1.

Corrected code to print values from 1 to 5 -

public class OffByOneDoWhileLoop {

public static void main(String[] args) {

int num = 1;

do {

System.out.println(num);

num++;

} while (num <6);

}

}

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?

Solution-- – this code doesn’t run infinitely .It prints three values that are- 0,2,4.

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?

Solution-- – This code generates error –“ incompatible types: int cannot be converted to Boolean” as the expression in while() is assignment rather than conditional.

The loop doesn’t run indefinitely.

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?

Solution--—This code prints three values – 0,2,4.

To print values from 0 to 4—

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

Solution--—The above code causes error because ‘x’ is declared within for() loop hence its scope is limited to the for() loop only.When accessed outside for9) it generates error.

Section 2

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.

Solution—

(i) (j)

a. 1 1 ---- print 1 1

1 2------ print 1 2

Cursor moved to next line

b. 2 1 -----print 2 1

2 2 -----print 2 2

Cursor moved to next line

c. 3 1 -----print 3 1

3 2 -----print 3 2

Cursor moved to next line

d. 4

end of for()

end of program.

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.

Solution—

total= 0

1. total
2. 5 5, 4 -- total value -4
3. 4 8, 7 – total value -7
4. 3 10 – total value- 10
5. 2 12, 11 – total value -11
6. 1 12, 11 – total value- 11
7. 0

End of for()

Print—11

End of program.

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.

Solution—

Count

1. 0

Inside while ()

1. 0 ----print 0

1

1. 1 –print 1

2

1. 2 – print 2

3

Out of while()

print –3

end of program.

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.

Solution---

(i)

a. 1

inside dowhile()

b. 1 --print--1

2

c. 2 ---print—2

3

d. 3—print--3

4

e.. 4 –print –4

5

Out of dowhile()

Print -5

End of program.

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.

Solution—

num (i)

a. 1

for ()

b. 1 1

0 1

c. 0 2

2 2

d. 2 3

-1 3

e. -1 4

3 4

f. 3 5

Out of for()

Print—3

End of program .

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.

Solution—

x ++x x-- --x x++ y

1. 5
2. 6 6 4 4 ------ (6-6+4+4)
3. 8

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.

Solution –

a b

10 5

++a b-- --a b++

11 5 10 4 (11\*5-10+4)

result

49

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.

Solution—

count (i) i++ ++i

1. 0

for()

1. 0 0
2. -2 0 2
3. -4 3 3 5
4. 6

Out of for()

Print— -4

Final output— -4

Section 3

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

Sol-

public class NaturalNoSum{

public static void main(String[] args) {

int sum=0;

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

{

sum+=i;

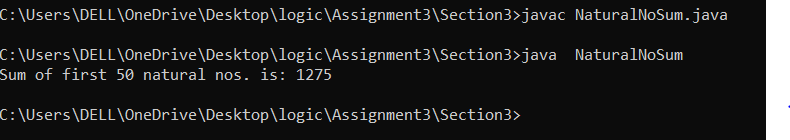
}

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

}

}

Output-



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

public class Factorial{

public static void main(String[] args) {

int n=10;

int fact=1;

while(n>0){

fact=fact\*n;

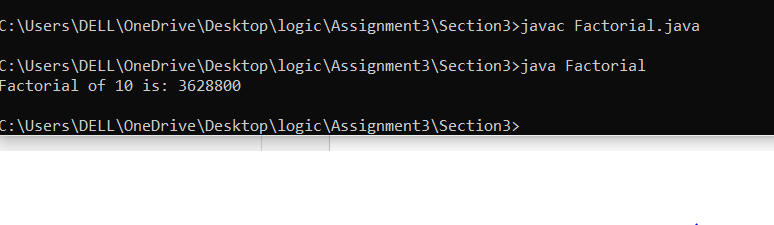
n--;

}

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

}

}



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

Solution—

public class MultipleOf7{

public static void main(String[] args) {

System.out.println("Multiples of 7 between 1 and 100 are: ");

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

if(i%7==0)

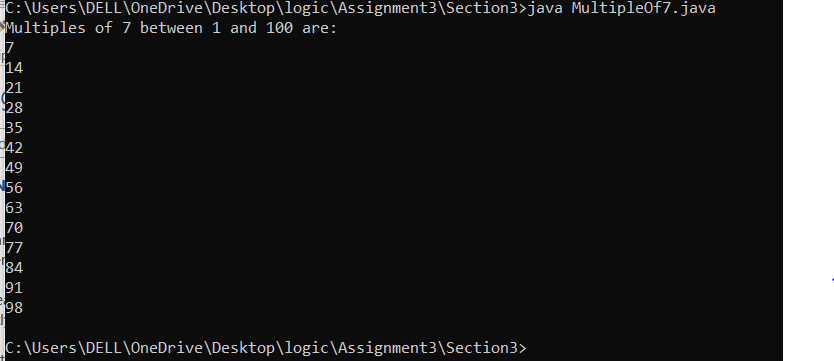
{

System.out.println(i);

}

}

}}



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

public class Reverse{

public static void main(String[] args) {

int n=1234;

int rev=0;

while(n>0){

int rem= n%10;

n=n/10;

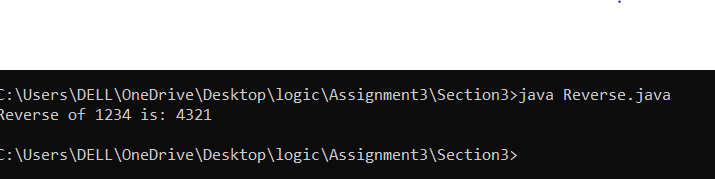
rev=rev\*10+rem;

}

System.out.println("Reverse of 1234 is: "+rev);

}

}



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

Solution—

public class Fibonacci{

public static void main(String[] args) {

int f=0;

int n=1;

System.out.print("Fibonacci series: ");

while(f<=21){

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

int a=f;

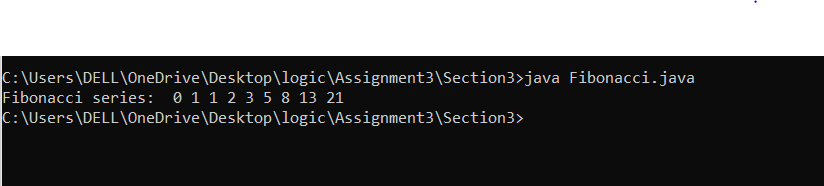
f=a+n;

n=a;

}

}

}



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

Solution—

import java.io.\*;

import java.util.\*;

public class PrimeNo{

public static void main(String[] args) {

int count=0;

System.out.println("Enter a number: ");

Scanner sc =new Scanner(System.in);

int n=sc.nextInt();

if(n==0||n==1||n==2)

{

System.out.println("It is a prime number ");

}

else {

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

{ if(n%i==0)

count++;

}

if(count>1)

{

System.out.println("It is not a prime number ");

}

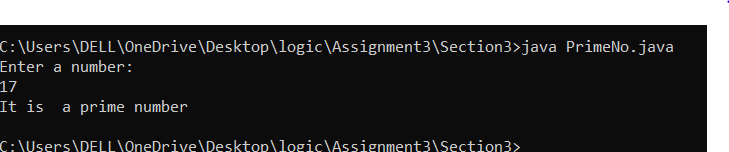
else

System.out.println("It is a prime 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).

Solution—

public class Sum{

public static void main(String[] args) {

int n=9876;

int sum=0;

while(n>0){

int rem=n%10;

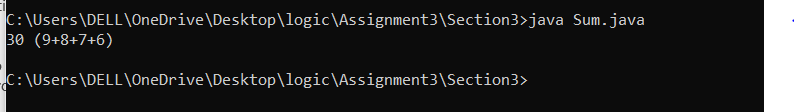
n=n/10;

sum = sum+rem;

}

System.out.println(sum+ " (9+8+7+6)");

} }



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

Solution—

public class CountDown{

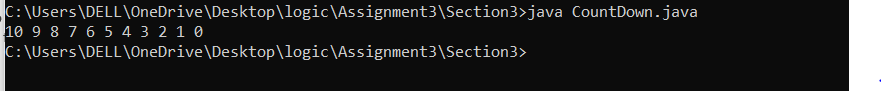
public static void main(String[] args) {

for(int i=10; i>=0;i--)

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

}

}



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

Solution—

public class LargestDigit{

public static void main(String[] args) {

int n=4825;

int l=0;

while(n>0){

int rem=n%10;

n=n/10;

if (rem>l)

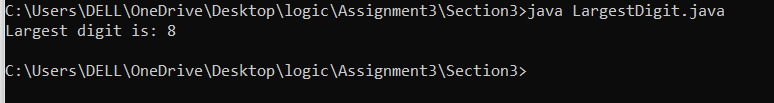
l=rem;

}

System.out.println("Largest digit is: "+l);

}

}



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

Solution—

public class EvenNo{

public static void main(String[] args) {

System.out.print("Even nos between 1 to 50 are: ");

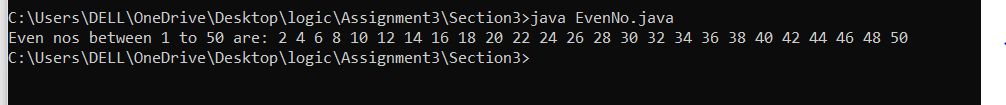
for(int i=1; i<=50;i++)

{ 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

Solution—

public class Incredecre{

public static void main(String[] args) {

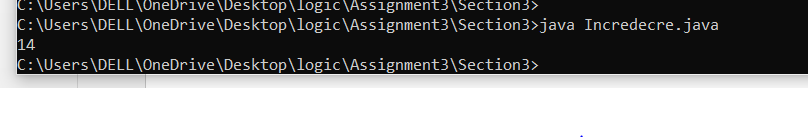
int a=5;

int b=6;

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

System.out.print(c);

}}



12. Write a program to draw the following pattern:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

Solution—

public class Q12{

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

Solution—

public class Q13{

public static void main(String[] args) {

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

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

{

if(i==1 || j==i )

System.out.print(i);

else

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

}

System.out.println();

}

for(int i=4;i>=1;i--){

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

{

if(i==1 || j==i )

System.out.print(i);

else

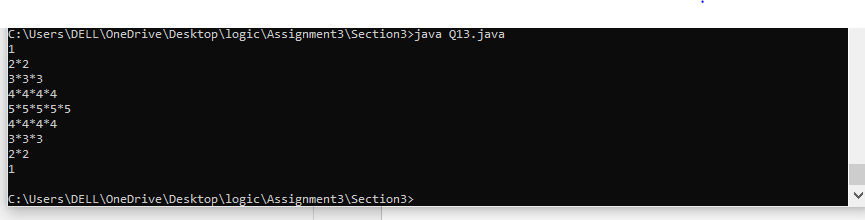
System.out.print(i+"\*");

}

System.out.println();

}

} }



14. Write a program to print the following pattern:

\*

\*\*

\*\*\*

\*\*\*\*\*

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

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

Solution—

15. Write a program to print the following pattern:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Solution—

public class Q15{

public static void main(String[] args) {

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

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

System.out.print(" ");

}

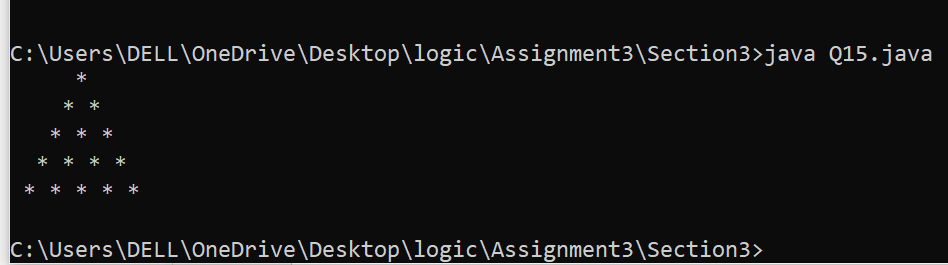
for (int k = 0; k <= i; k++) {

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

}

System.out.println();

} } }



16. Write a program to print the following pattern:

\*

\*\*\*

\*\*\*\*\*

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

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

Solution—

public class Q16{

public static void main(String[] args) {

int n=2;

for(int i=6;i>1;i--){

for(int k=1;i-k>1;k++)

{

System.out.print(" ");

}

if(n<11){

for(int j=1;n-j>0;j++)

{

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

}

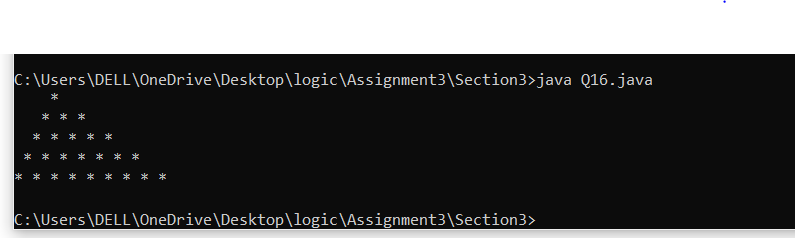
}

n+=2;

System.out.println();

}

} }



17. Write a program to print the following pattern:

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Solution—

public class Q17{

public static void main(String[] args) {

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

{

for(int k=5; k-i>0;k--)

{

System.out.print(" ");

}

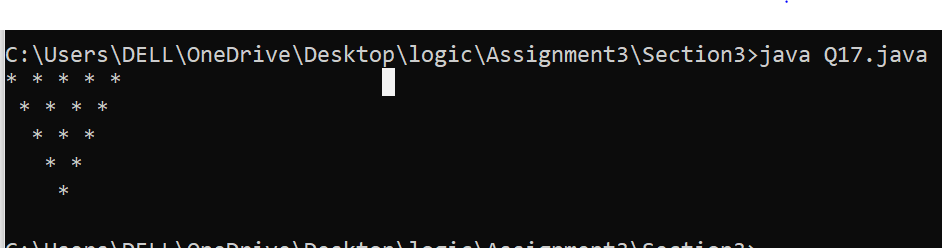
for (int j=1;j<=i;j++)

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

}

System.out.println();

} }}



18. Write a program to print the following pattern:

\*

\*\*\*

\*\*\*\*\*

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

\*\*\*\*\*

\*\*\*

\*

Solution—

public class Q18{

public static void main(String[] args) {

int n=2;

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

for(int k=1;i-k>1;k++)

{

System.out.print(" ");

}

if(n<9){

for(int j=1;n-j>0;j++)

{

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

}

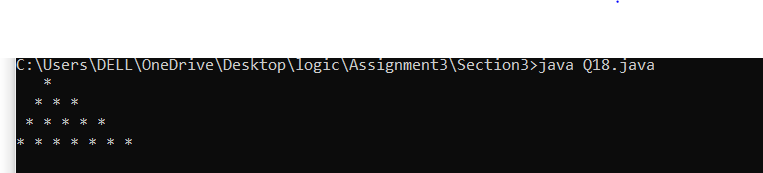
}

n+=2;

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 Q19{

public static void main(String[] args) {

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

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

{

if(j<i)

{ System.out.print(j+"\*");

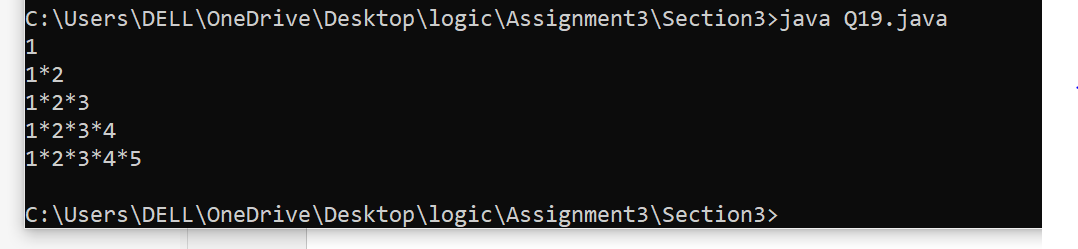
}

}

System.out.println(i);

}

} }



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 Q20{

public static void main(String[] args) {

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

for(int j=5;j>=i;j--)

{

if(j>i)

{

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

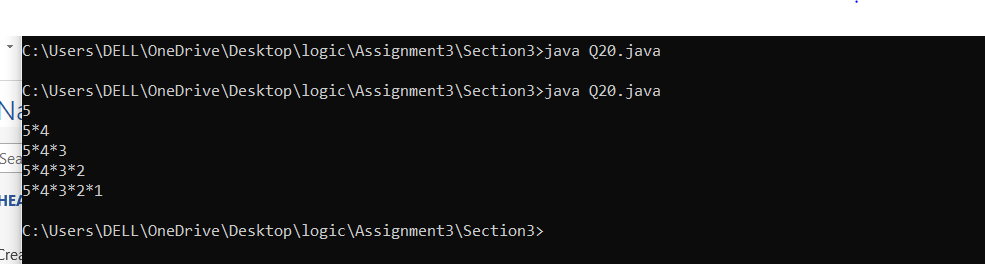
}

}

System.out.println(i);

}

} }



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—

public class Q21{

public static void main(String[] args) {

for(int i=1;i<=9;i+=2){

{

for(int j=1;j<=i;j+=2)

{

if(i>j)

{

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

}

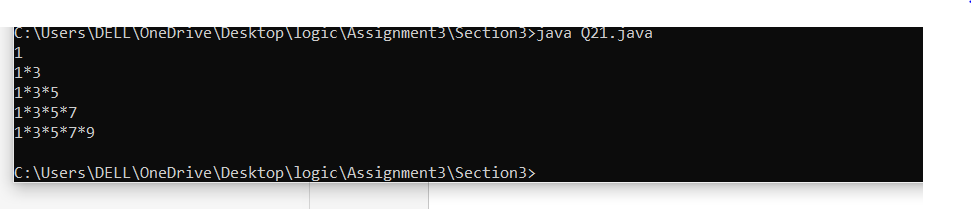
}

System.out.println(i);

}

}

} }



22. Write a program to print the following pattern:

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

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

\*\*\*\*\*

\*\*\*

\*

\*\*\*

\*\*\*\*\*

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

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

Solution—

public class Q22{

public static void main(String[] args) {

int n1=0;

for(int i=1;i<4;i++){

for(int k=1;k-i>1;k++)

{

System.out.print(" ");

}

if(n1<=9){

for(int j=9;n1-j>0;j--)

{

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

}

}

n1+=2;

System.out.println();

}

int n2=2;

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

for(int k=1;i-k>1;k++)

{

System.out.print(" ");

}

if(n2<11){

for(int j=1;n2-j>0;j++)

{

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

}

}

n2+=2;

System.out.println();

}

} }

23. Write a program to print the following pattern:

11111

22222

33333

44444

55555

Solution—

class Q23{

public static void main(String[] args) {

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

for(int j=1;j<=5;j++)

{

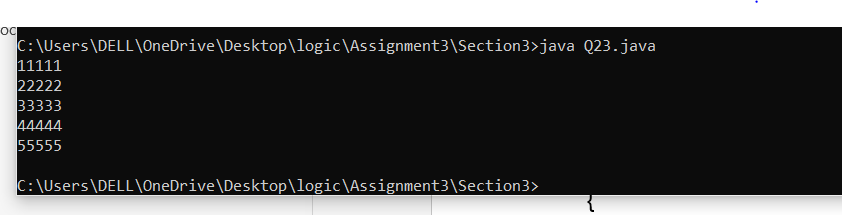
System.out.print (i);

}

System.out.println ();

}

} }



24. Write a program to print the following pattern:

1

22

333

4444

55555

Solution—

public class Q24{

public static void main(String[] args) {

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

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

{

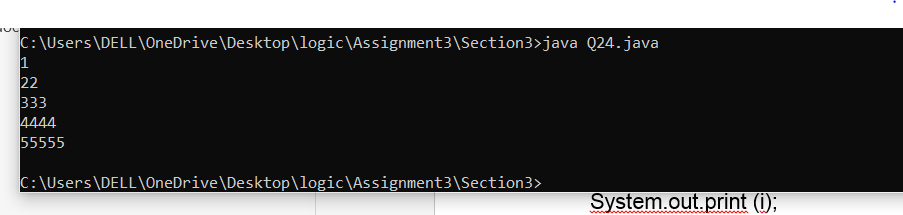
System.out.print (i);

}

System.out.println ();

}

} }



25. Write a program to print the following pattern:

1

12

123

1234

12345

Solution—

public class Q25{

public static void main(String[] args) {

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

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

{

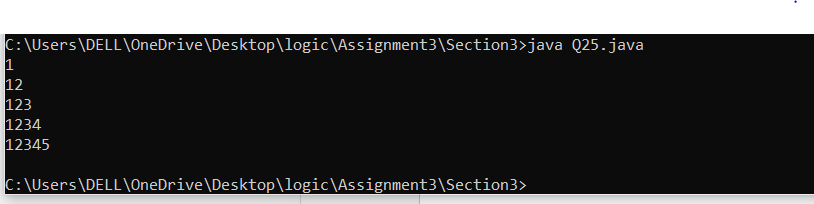
System.out.print (j);

}

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

Solution—

public class Q25{

public static void main(String[] args) {

int a=1;

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

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

{

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

a++;

}

System.out.println();

}

} }

