2.7)

A. Comment

B. Condition

C. division and remainder

D. Innermost

E. Variable

2.8)

A) System.out.print(“Enter an integer.”);

B) Int a = b\*c

C) //This program performs a sample payroll calculation

2.9)

A. True. It is executed first because it is a nested parenthesis.

B. False. Variable identifiers should not start with a capital letter.

C. True. 2+3+5 will be seen as a single value to be divided by the denominator 4

D. False. Variable identifiers can start with lower case letters.

2.10)

A) x=10

B)Value of 5\*1 is 5

C) X is 5 and y is 1

D) 6 is not equal to 5

2.11)

A,C and D contain variables who do not have modified values .

2.12)

A, B and D statements are correct

2.13)

The output will be 1314 and the cursor will remain on the same line. The two expressions will display on the screen. They have format specifiers that hold a value (p+2\*4) and (p\*2+4) respectively. The value of the first statement is displayed and immediately followed by the second because there is no specification for spaces in between or display on separate lines.

2.14)

import java.util.Scanner;

public class A{

public static void main(String[] args){

Scanner userInput = new Scanner(System.in);

System.out.println("Enter numbers 1 to 4 separated with spaces: ");

int a = userInput.nextInt();

int b = userInput.nextInt();

int c = userInput.nextInt();

int d = userInput.nextInt();

System.out.println( a + " "+ b +" "+ c + " "+ d);

}

}

b)

import java.util.Scanner;

public class B{

public static void main(String[] args){

Scanner userInput = new Scanner(System.in);

System.out.println("Enter numbers 1 to 4 separated with spaces: ");

int a = userInput.nextInt();

int b = userInput.nextInt();

int c = userInput.nextInt();

int d = userInput.nextInt();

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

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

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

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

}

}

C)

import java.util.Scanner;

public class C{

public static void main(String[] args){

Scanner userInput = new Scanner(System.in);

System.out.println("Enter numbers 1 to 4 separated with spaces: ");

int a = userInput.nextInt();

int b = userInput.nextInt();

int c = userInput.nextInt();

int d = userInput.nextInt();

System.out.printf("%d %d %d %d\n", a, b, c, d);

}

}

2.15)

import java.util.Scanner;

public class A{

public static void main(String[] args){

Scanner userInput = new Scanner(System.in);

System.out.print("Enter first integer: ");

int a = userInput.nextInt();

System.out.print("Enter second integer: ");

int b = userInput.nextInt();

int squareA = a \* a;

int squareB = b \* b;

int sumOfSquares = squareA + squareB;

int differenceOfSquares = squareA - squareB;

System.out.printf("Square of first and second integer each %d %d %n", squareA, squareB);

System.out.printf("Sum of the Square of first and second integer %d %n", sumOfSquares);

System.out.printf("Difference of the Square of first and second integer %d %n", differenceOfSquares);

}

}

2.16)

import java.util.Scanner;

public class CompareIntegers{

public static void main(String[] args){

Scanner userInput = new Scanner(System.in);

System.out.print("Enter an integer: ");

int integer = userInput.nextInt();

int integerSquare = integer \* integer;

if (integer > 100){

System.out.printf("%d > 100 %n", integer);

}

if (integer == 100){

System.out.printf("%d == 100 %n", integer);

}

if (integer != 100){

System.out.printf("%d != 100 %n", integer);

}

if (integer < 100){

System.out.printf("%d < 100 %n", integer);

}

if (integerSquare > 100){

System.out.printf("The square of integer %d > 100 %n", integerSquare);

}

if (integerSquare == 100){

System.out.printf("The square of integer %d == 100 %n", integerSquare);

}

if (integerSquare != 100){

System.out.printf("The square of integer %d != 100 %n", integerSquare);

}

if (integerSquare < 100){

System.out.printf("The square of integer %d < 100 %n", integerSquare);

}

}

}

2.19)

\*\*\*\*

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

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

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

\*\*\*\*

2.20)

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*

2.21)

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

2.22)

\*\*\*\*

\*\*\*\*\*

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

2.23)

\*

\*\*\*

\*\*\*\*\*

\*\*\*