**Exersise for Language Fundamentals**

1.What do you know about Java?

2.Why Java is platform independent?

3.What are the supported platforms by Java Programming Language?

4.List two Java IDE’s?

5.What is the Difference between JDK and JRE ?

6.Why people say that Java is 'write once and run anywhere' language?

7.Insert the missing part of the code below to output "Hello World".

public class MyClass {

public static void main(String[] args) {

. . ("Hello World");

}

}

8.Comments in Java are written with special characters. Insert the missing parts:

...This is a single-line comment

...This is a multi-line comment...

9.Write a Java program to print 'Hello' on screen and then print your name on a separate line.

Expected Output:

Hello

Alexandra Abramov

10.Write a Java program to print the sum of two numbers.

Test Data:

74 + 36

Expected Output :

110

11.Write a Java program to divide two numbers and print on the screen.

Test Data :

50/3

Expected Output :

16

12. Write a Java program to print the result of the following operations. Go to the editor

Test Data:

a. -5 + 8 \* 6

b. (55+9) % 9

c. 20 + -3\*5 / 8

d. 5 + 15 / 3 \* 2 - 8 % 3

Expected Output :

43

1

19

13.Write a Java program that takes two numbers as input and display the product of two numbers.

Test Data:

Input first number: 25

Input second number: 5

Expected Output :

25 x 5 = 125

14. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers. Go to the editor

Test Data:

Input first number: 125

Input second number: 24

Expected Output :

125 + 24 = 149

125 - 24 = 101

125 x 24 = 3000

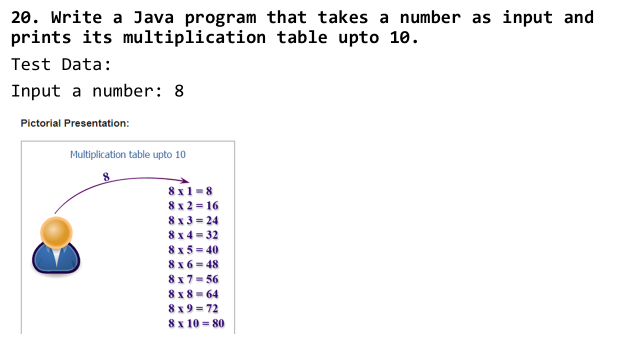
125 / 24 = 5

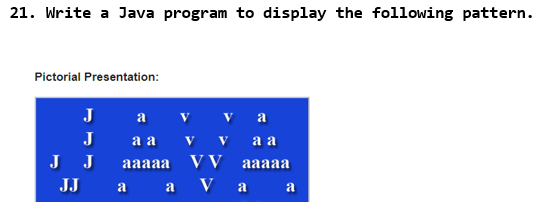
125 mod 24 = 5

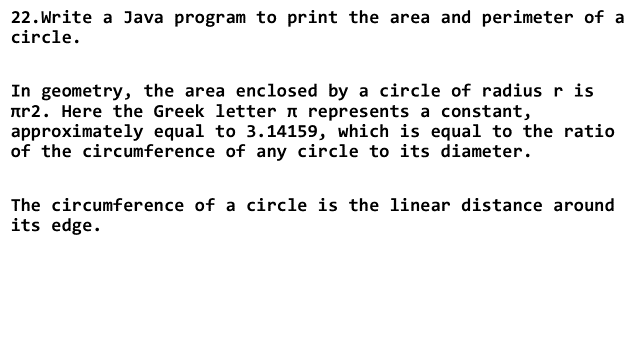
-----------------------------------------------------

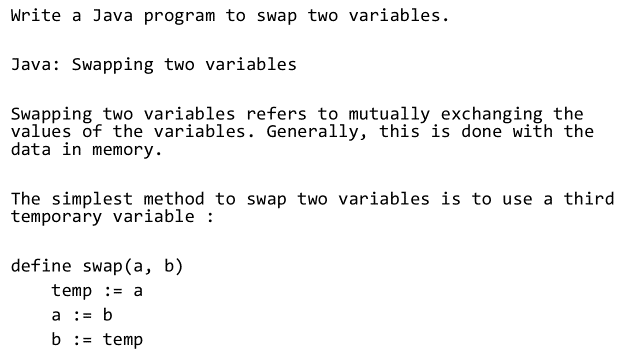
20.04.21

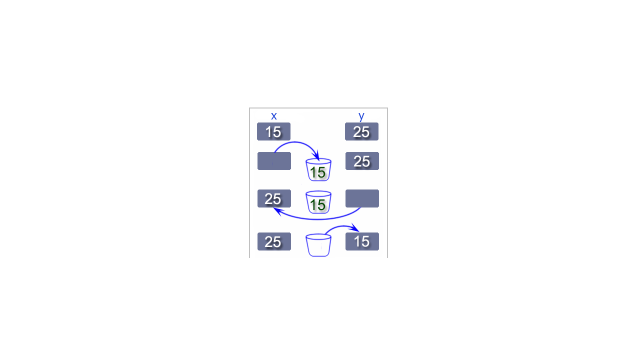
1. LF/app7/src/ - N,O

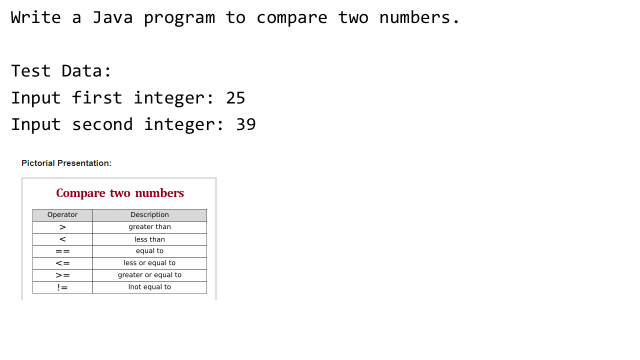


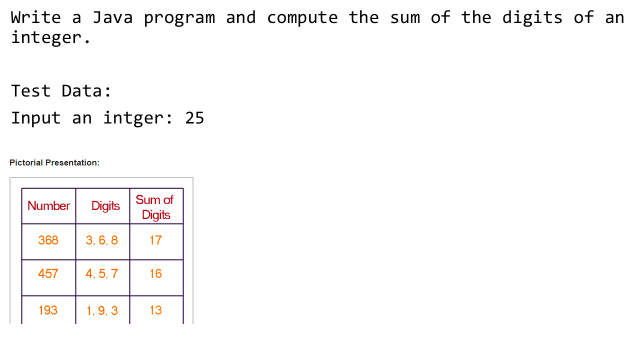


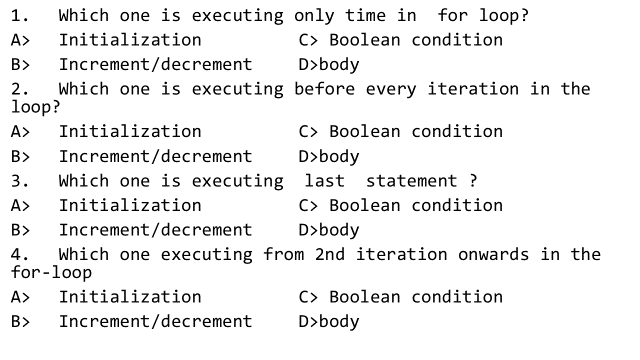


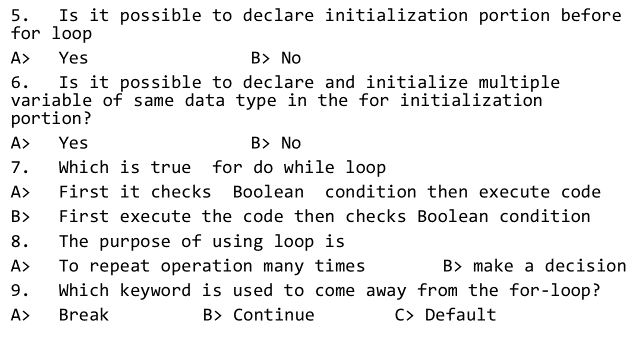


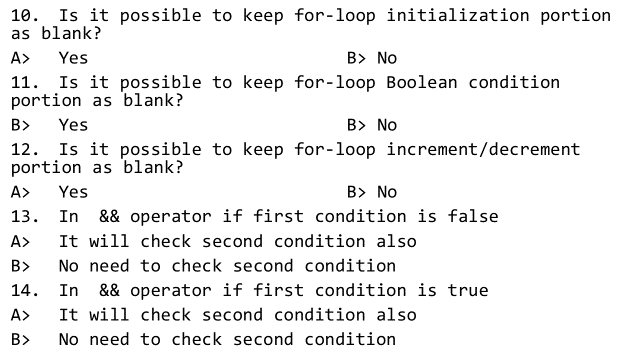


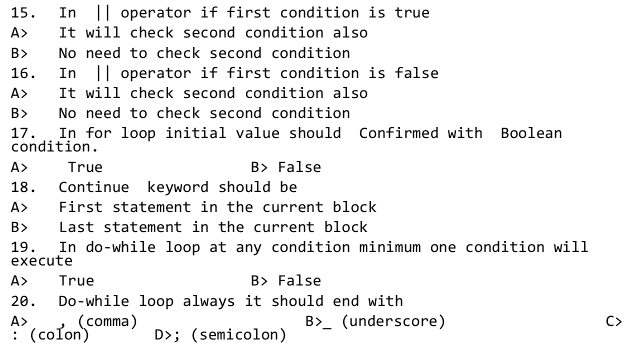


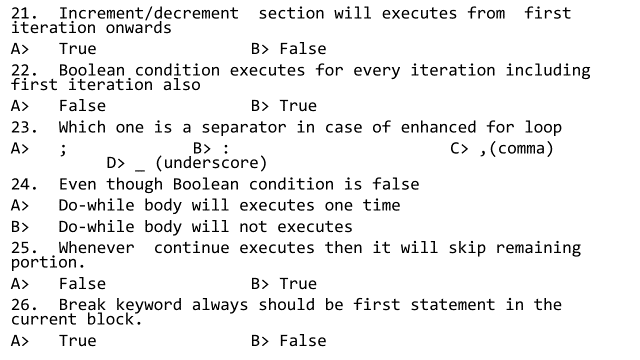
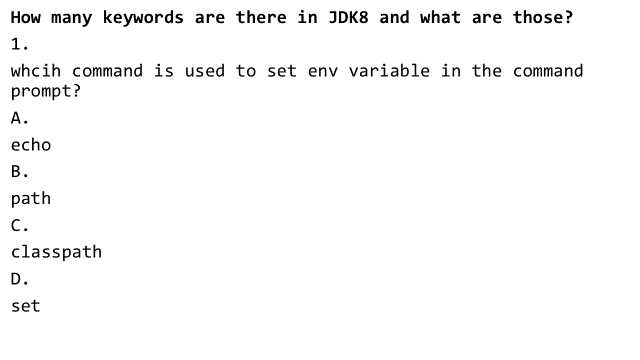


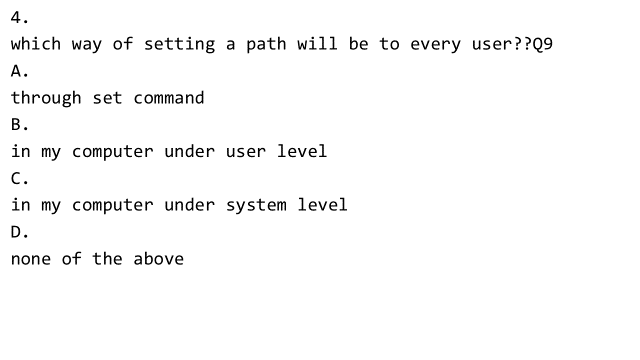




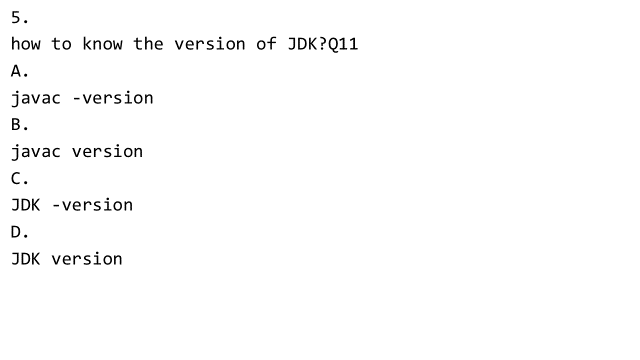
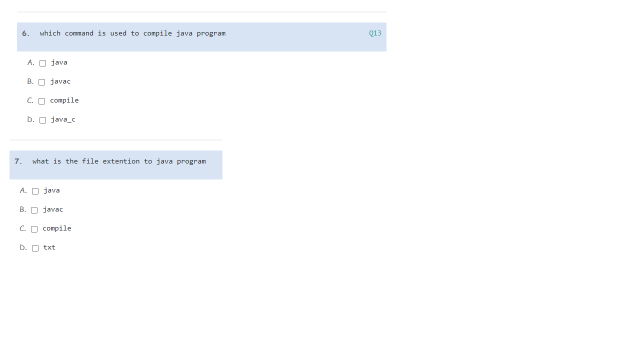


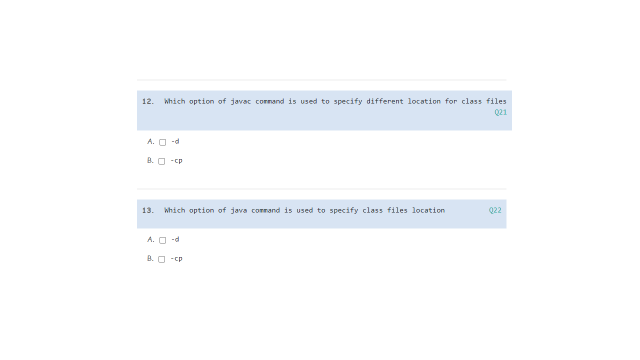


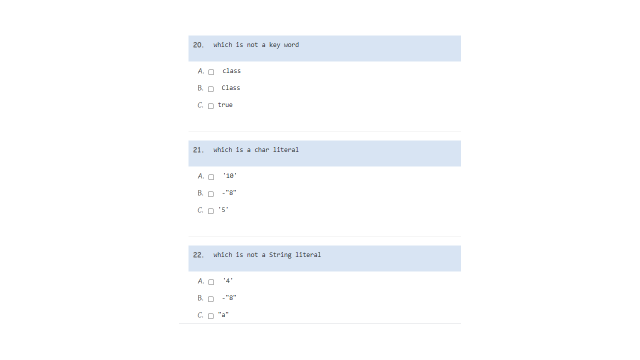




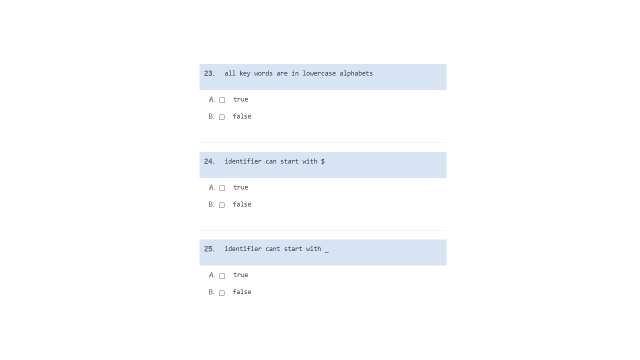


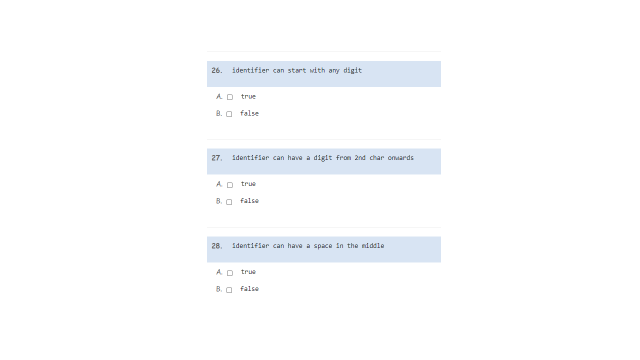


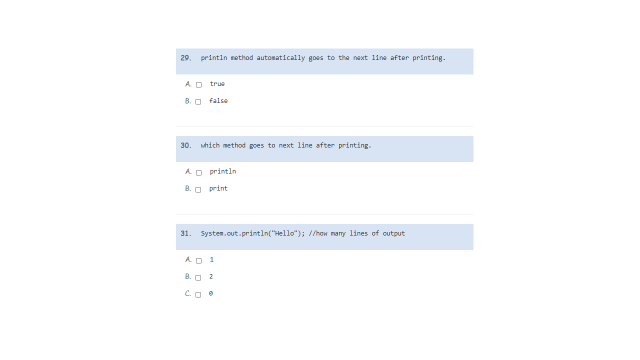


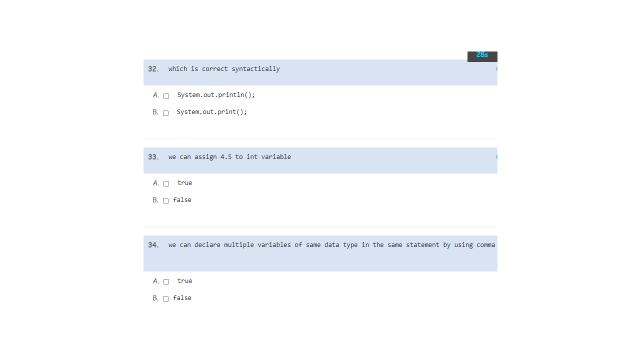






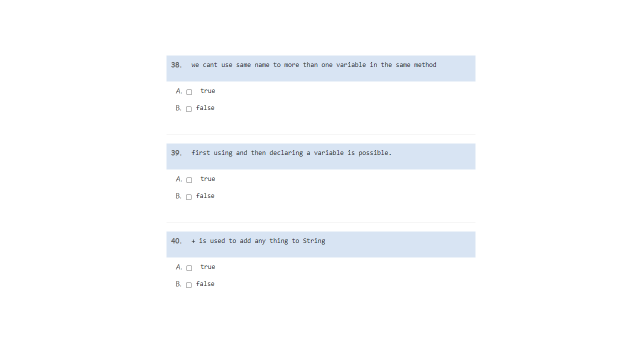












**Assignment for 21.04.21**

How many times 'Hello' is printed?

public class CppBuzz {

public static void main(String[] args){

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

{

System.out.println("Hello");

i++;

i--;

}

}

}

How many times 'Hello' is printed?

public class CppBuzz {

public static void main(String[] args){

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

{

System.out.println("Hello");

i+=2;

}

}

}

public class CppBuzz {

public static void main(String[] args){

for(int i = 0; i<5; i+=2)

{

System.out.println("Hello");

i+=2;

}

}

}

How many times 'cppbuzz' is printed?

public class Main

{

public static void main(String[] args) {

while(true){

System.out.println("cppbuzz");

}

}

}

How many times 'cppbuzz' is printed?

public class Main

{

public static void main(String[] args) {

do{

System.out.println("cppbuzz");

}while(false);

}

}

Write a method that uses a for loop to display numbers in descending order in the step of 5 starting from 100. Call your method from the main method.

e.g., The output should be:

100

95

90

...

...

5

0

2) Choose a valid loop name in Java below.

A) for

B) while

C) do while

D) All

3) Every loop in Java has a condition that should be \_\_\_ in order to proceed for execution. (TRUE / FALSE)

A) FALSE

B) TRUE

4) Choose the correct syntax of the WHILE loop in Java below.

A)

while(condition)

{

//statements

}

B)

while(condition);

{

//statements

}

C)

while

{

//statements

}(condition)

D) None

5) Choose the correct Syntax of FOR loop in Java below.

A)

for(initialization; condition; increment-or-decrement)

{

//statements

}

B)

for(condition; increment-or-decrement; initialization)

{

//statements

}

C)

for(increment-or-decrement; condition; initialization)

{

//statements

}

D) None

6) Choose the correct syntax of the DO WHILE loop in Java below.

A)

do

{

//statements

}while(condition);

B)

do

{

//statements

}while(condition)

C)

do while(condition)

{

//statements

}

D) None

7) Choose the correct syntax of an Enhanced FOR loop in Java below.

A)

for(Type variable: Collection)

{

//statements

}

B)

for(Type variable; Collection)

{

//statements

}

C)

for(Collection: Type variable)

{

//statements

}

D) None

8) State TRUE or FALSE. A WHILE loop in Java executes the statements at least once even the condition is not satisfied.

A) FALSE

B) TRUE

9) A BREAK statement inside a Loop like WHILE, FOR, DO WHILE and Enhanced-FOR causes the program execution \_\_\_ Loop.

A) Exit

B) Continuation with next iteration

C) Never exit

D) None

10) A CONTINUE statement inside a Loop like WHILE, FOR, DO-WHILE and Enhanced-FOR causes the program execution \_\_\_ the loop.

A) Skip

B) Skip present iteration and continue with next iteration of the loop

C) Exit

D) None

11) Choose the Java-Code below with a never-ending loop.

A)

while(true);

B)

for(;true;);

C)

do

{

;

}while(true);

D) All

12) A loop in Java generally contains a Loop-Counter variable. State TRUE or FALSE.

A) FALSE

B) TRUE

C) -

D) -

13) An Increment operator "++" and/or a Decrement operator "--" are used along with a Loop-Counter variable in Java. (TRUE / FALSE).

A) FALSE

B) TRUE

C) -

D) -

14) What is the output of the below Java program?

int a=1;

while(a<4)

{

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

a++;

}

A) 1 2 3 4

B) 1 2 3

C) 6

D) Compiler error

15) What is the output of the below Java program with a decrement operator and WHILE-loop?

int a=4;

while(a>0)

{

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

a--;

}

A) 4 3 2 1

B) 3 2 1

C) Compiler error

D) None

16) What is the output of the below Java program?

String str="FOX";

int i=0;

while(i<str.length())

{

System.out.print(str.charAt(i));

i++;

}

A) FFF

B) FOX

C) Compiler error

D) None

17) What is the output of the below Java program with WHILE, BREAK and CONTINUE?

int cnt=0;

while(true)

{

if(cnt > 4)

break;

if(cnt==0)

{

cnt++;

continue;

}

System.out.print(cnt + ",");

cnt++;

}

A) 0,1,2,3,4,

B) 1,2,3,4,

C) 1,2,3,4

D) Compiler error

18) What is the main difference between a WHILE and a DO-WHILE loop in Java?

A) WHILE loop executes the statements inside of it at least once even if the condition is false.

B) DO-WHILE loop executes the statements inside of it at least once even if the condition is false.

C) WHILE loop is fast.

D) DO-WHILE loop is fast.

19) What is the value of "age" in the below Java program with a DO-WHILE loop?

int age=20;

do

{

age++;

}while(age<20);

System.out.println(age);

A) 20

B) 21

C) Compiler error

D) None

20) What is the output of the below java program that implements nesting of loops?

int i=1, j=1;

while(i<3)

{

do

{

System.out.print(j + ",");

j++;

}while(j<4);

i++;

}

A) 1,2,3,4,1,2,3,4,

B) 1,2,3,4,

C) 1,2,3,1,2,3,

D) 1,2,3,

21) What is the output of the below Java program?

int time=50;

do

{

System.out.print(time + ",");

time++;

}while(time < 53)

A) 50,50,50,

B) 50,51,52,

C) 51,52,53,

D) Compiler error

22) What is the output of the below Java program?

char ch[] = {'A', 'B', 'C'};

int i=0;

do

{

System.out.print(ch[i] + ",");

i++;

}while(i < ch.length);

A) A,B,C,

B) A,B,C

C) A,A,A

D) Compiler error

23) What is the output of the below Java program?

String str[] = {"A","B","C"};

int i=0;

do

{

if(i>= str.length)

break;

System.out.print(str[i] + ",");

i++;

}while(true);

A) A,B,C,

B) A,B,C

C) Runtime Exception with Index Of Bounds Exception

D) Compiler error

24) What is the output of the below Java code with a FOR loop?

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

{

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

}

A) 1,2,3,4,

B) 1,2,3,4

C) 1,2,3,4,5,

D) 1,2,3,4,5

25) What is the output of the below Java code?

boolean[] ary = {true, false, true, true};

for(int i=0; i<ary.length; i++)

{

System.out.print(ary[i] +",");

}

A) true,true,true,true,

B) true,false,false,true

C) true,false,true,true

D) Compiler error

26) What is the output of the below Java code?

int score=1;

for(; true; score++)

{

System.out.print(score +",");

if(score > 3)

break;

}

A) 1,2,3,

B) 1,2,3

C) 1,2,3,4,

D) 1,2,3,4

27) What is the output of the below Java program with FOR loop?

for(int j=0; j<5;j++;)

System.out.print(j + ",");

A) 1,2,3,4,

B) 0,1,2,3,4

C) Compiler error

D) None

28) State TRUE or FALSE. In a FOR loop, the Initialization-part, Condition-part and Increment/Decrement part can be empty.

A) FALSE

B) TRUE

C) -

D) -

29) Any loop can be nested inside any loop in Java. (TRUE/FALSE).

A) FALSE

B) TRUE

C) -

D) -

30) A Loop in Java language may contain \_\_\_.

A) Any loop

B) IF-ELSE statements

C) SWITCH statements

D) All

31) In Java language, BREAK or CONTINUE statements can be implemented inside a Loop only with the help of \_\_\_ statements to avoid never-ending loops.

A) IF ELSE

B) SWITCH

C) ENUM

D) None

32) The Enhanced FOR loop in Java was introduced by \_\_\_.

A) JDK 4

B) JDK 5

C) JDK 6

D) JDK 7

33) An enhanced FOR loop work with only Collection type data. Examples of Collection are \_\_\_.

A) Array Class type or any regular array variable

B) ArrayList

C) HashMap, HashSet

D) All

34) What is the output of Java Enhanced FOR loop below?

String names[] = {"MOGLI", "SHAREKHAN", "BALU"};

for(String str: names)

{

System.out.print(str + ",");

}

A) MOGLI,

B) MOGLI,SHAREKHAN,

C) MOGLI,SHAREKHAN,BALU,

D) Compiler error

35) An Enhanced FOR loop in Java misses \_\_\_ and \_\_ compared to the old-style FOR loop.

A) Speed and Easiness

B) Initialization, Increment/Decrement

C) Semicolons, Variables

D) None

36) What is the output of the Java program with Enhanced FOR loop below?

String countries[] = {"BRAZIL", "CHILE", "SYDNEY"};

int i=0;

for(String str: countries)

{

if(i<2)

;

else

break;

System.out.print(str + ",");

i++;

}

A) BRAZIL,CHILE,SYDNEY,

B) BRAZIL,CHILE,

C) BRAZIL,

D) Compiler error

37) What is the output of the Java code snippet?

int i=0;

for(i=1; i<=6;i++)

{

if(i%3==0)

continue;

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

}

A) 1,2,

B) 1,2,4,5,

C) 3,6,

D) Compiler error

38) A BREAK or CONTINUE statement applies only to the \_\_\_ loop.

A) Inner loop or the loop containing break or continue

B) always Outer loop

C) Sometimes inner loop, sometimes outer loop

D) None

39) A BREAK-WITH-LABEL or CONTINUE-WITH-LABEL are used in particular in Java to select \_\_ loop either to Break or Continue.

A) Inner loop

B) Outer loop

C) -

D) -

40) Choose rules for naming a Label in Java below.

A) The name of a label or identifier may start only with Alphabet, Underscore ( \_ ) or Dollar ($) symbol

B) A label is kept before the loop in general

C) Duplicate label names are not allowed

D) All

41) State TRUE or FALSE. You can exit an inner loop without using a BREAK statement but with a CONTINUE and Label on the outer loop.

A) FALSE

B) TRUE

C) -

D) -

42) The keyword "goto" can be used in Java programs with labels. (TRUE/FALSE)

A) FALSE

B) TRUE

C) -

D) -

43) Is it possible to break all loops with a single BREAK with a Label statement? (YES/NO)

A) YES

B) NO

C) -

D) -

44) What is the output of the Java code snippet below?

outer:

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

{

inner:

for(int j=1; j<=4;j++)

{

if(j==1)

break outer;

}

System.out.print("A");

}

A) A

B) AAAA

C) No Output

D) Compiler error

45) What is the output of the below Java program?

outer:

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

{

inner:

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

{

if(j>i)

break inner;

System.out.print(j +",");

}

}

A) 1,1,1

B) 1,2,2,

C) 1,1,2,

D) Compiler error

What will be the output for the following program?

class Test {

public

static void main(String[] args)

{

do

while (true)

System.out.println("HELLO");

while (false);

}

}

What will be the output for the following program?

class Test {

public

static void main(String[] args)

{

do

System.out.println("FRIENDS");

while (true);

System.out.println("ENEMY");

}

}

Options:

1. Compile time error

2. FRIENDS

3. No output

4. ENEMY

What will be the output for the following program?

class Test {

public

static void main(String[] args)

{

int x = 1, y = 2;

do

System.out.println("FRIENDS");

while (x < y);

System.out.println("ENEMY");

}

}

Options:

1. FRIENDS

2. ENEMY

3. No Output

4. FRIENDS (Infinitely)

What will be the output for the following program?

class Test {

public

static void main(String[] args)

{

do

while (true)

;

System.out.println("HELLO");

}

}

What will be the output for the following program?

class Test {

public static void main(String[] args)

{

do {

System.out.print(1);

do {

System.out.print(2);

} while (false);

} while (false);

}

}

Options:

1. 12

2. 21

3. 1

4. 2

What will be the output of the following program?

class ForSample

{

public static void main(String s[])

{

int i = 0;

for(;i <= 5; i++ )

{

System.out.println("i = " + i );

}

System.out.println("i after the loop = " + i );

}

}

What will be the output of the following program?

class Factorial

{

public static void main(String s[])

{

int number = 5;

int factorial = 1;

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

{

factorial \*= factorial;

}

System.out.println("Factorial of 5 is " + factorial);

}

}

What will be the output of the following program?

class MultiVariableFor

{

public static void main(String s[])

{

int a, b;

for(a = 1, b = 4; a < b; a++, b--)

{

System.out.println("a = " + a);

System.out.println("b = " + b);

}

}

}

What will be the output of the following program?

class OutPut

{

public static void main(String s[])

{

int i = 5;

for(; i < 6 || i >= 0; )

switch(i)

{

case 0:

System.out.println("i is zero.");

i++;

break;

case 1:

System.out.println("i is one.");

i++;

break;

case 2:

System.out.println("i is two.");

i++;

continue;

case 3:

System.out.println("i is three.");

i++;

break;

default:

System.out.println("i is greater than three.");

i -= 6;

}

}

}

What will be the output of the following program?

class OutPut

{

public static void main(String s[])

{

int i = 0;

for(; i < 6 && i >= 0; )

switch(i)

{

case 0:

System.out.println("i is zero.");

i++;

break;

case 1:

System.out.println("i is one.");

i++;

break;

case 2:

System.out.println("i is two.");

i++;

continue;

case 3:

System.out.println("i is three.");

i++;

break;

default:

System.out.println("i is greater than three.");

i -= 6;

}

}

}

What will be the output of the following program?

public class NewQuestion

{

public static void main(String[] args)

{

int j = 6, i = 0;

System.out.print("Now i value and j value: ");

for (int x = 1; i < ++x; i += x, j -= x)

{

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

}

}

}

What will be the output of the following program?

public class TestProgram

{

public static void main(String[] args)

{

int j = 6, i = 0;

System.out.print("Now i value and j value: ");

for (int x = 0; i < ++x + x++; i += x, j -= x)

{

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

x--;

}

}

}

What will be the output of the following program?

public class Operation {

public static void main(String[] args) {

int num1 = 0;

int num2 = 0;

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

if ((++num1 > 2) && (++num2 > 2)) {

num1++;

}

}

System.out.println(num1 + " and " + num2);

System.out.println(num2 | 12);

}

}

What will be the output of the following program?

public class IsAPerfectNumber

{

public static void main(String args[])

{

int number = 496;

int sum = 0;

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

{

if (number % i == 0)

sum = sum + i;

}

System.out.print((sum == number) ? "Yes" : "No");

}

}

What will be the output of the following program?

public class Loops {

public static void main(String args[]) {

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

if (i % (3 \* 5) == 0)

System.out.print("Andhra" + ",");

else if (i % 5 == 0)

System.out.print("Pradesh" + ",");

else if (i % 3 == 0)

System.out.print("India" + ",");

else

System.out.println(i + ",");

}

}

}

What will be the output of the following program?

public class Mock {

public static void main(String args[]) {

int z = 3;

for (int i = 0; i < 2; i++) {

z++;

switch (z) {

case 3 :

System.out.print((z = z + 1) + " ");

case 5 :

System.out.print((z = z + 2) + " ");

break;

default :

System.out.print((z = z + 8) + " ");

case 6 :

System.out.print((z = z + 4) + " ");

}

z--;

}

}

}

What will be the output of the following program?

public class Mock {

public static void main(String args[]) {

int z = 3;

for (int i = 0; i < 2; i++) {

switch (z) {

case 3 :

System.out.print((z = z + 1) + " ");

case 5 :

System.out.print((z = z + 2) + " ");

break;

default :

System.out.print((z = z + 8) + " ");

case 6 :

System.out.print((z = z + 4) + " ");

}

}

}

}

What will be the output of the following program?

public class Mock {

public static void main(String args[]) {

int z = 3;

for (int i = 0; i < 2; i++) {

z++;

switch (z) {

case 3 :

System.out.print(z = z + 1 + " ");

case 5 :

System.out.print(z = z + 2 + " ");

break;

default :

System.out.print(z = z + 8 + " ");

case 6 :

System.out.print(z = z + 4 + " ");

}

z--;

}

}

}

What will be the output of the following program?

public class YZ {

public static void main(String args[]) {

int z = 0;

for (int y = 0; y >= z; ++y, z++) {

System.out.print(y + "~");

System.out.println(z);

}

}

What will be the output of the following program?

public class MyScope {

public static void main(String[] args) {

for (int i = 0; i < 3; i++)

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

System.out.println(i);

}

}

What will be the output of the following program?

public class LoopFor {

public static void main(String[] args) {

int i = 0;

for (; i < 3; i++)

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

System.out.println(i);

}

}

Compare the following programs ForSample1, ForSample2 and WhileSample

class ForSample1

{

public static void main(String s[])

{

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

{

System.out.println("i = " + i );

}

}

}

class ForSample2

{

public static void main(String s[])

{

int i = 0;

for( ;i <= 5; )

{

System.out.println("i = " + i );

i++;

}

}

}

class WhileSample

{

public static void main(String s[])

{

int i =0;

while(i <= 5)

{

System.out.println("i = " + i );

i++;

}

}

}

Output of all the three programs is same.

Output of ForSample1 and ForSample2 is same, but they are different from WhileSample.

Output of ForSample2 and WhileSample is same, but they are different from ForSample1.

ForSample2 does not compile since there is no code before first semicolon and no code after second semicolon.

What will be output of the following program?

class DoWhile

{

public static void main(String s[])

{

int n = 3;

do

{

System.out.println("n = " + n);

n--;

} while ( n > 0 );

}

}

What will be output of the following program?

class DoWhile

{

public static void main(String s[])

{

int n = 5;

do

{

System.out.println("n = " + n);

n--;

} while ( n < 2 );

}

}

What will be the output of the program?

class Output

{

public static void main(String [] args)

{

int I = 1;

do while ( I < 1 )

System.out.print("I is " + I);

while ( I > 1 ) ;

}

}

What will be the output of the following program?

public class MastiBrother {

public static void main(String[] args) {

int i = 5;

while (i < 7) {

for (int k = 0; k < 2; k++) {

System.out.print(i);

}

i++;

}

}

}

**22.04.21 Exercises**

### (1) What is static in java?

### (2) Why we use static in java?

### (3) Where we can use static keyword?

### (4) What is static variable in java?

### (5) What is static method in java?

### (6) What is static block?

### (7) Can we Override static method?

### (8) Can we Overload static method?

### (9) Why java main method is static?

### (10) Can we declare multiple static blocks in our code?

### (11) Can we call super class static method in sub class?

### (12) Can constructor be static in java?

### (13) Can we access non-static data member in static method in java?

### (14) Can we access static data member in static method?

### (15) Can we use this and super in static context?

### (16) Can we write static public void main(String args[])?

### (17) Can abstract class have static variable in it?

### (18) Can we execute java program without main() method?

Write the output for the following programs

class U

{

static int i = test1();

static int j = test2();

static

{

System.out.println("SIB:" + "i:" + i + ", " + "j:" + j);

i = i + 1;//i += 1

j = j + i;//j += i

main(null);

System.out.println("SIBAgain:" + "i:" + i + "," + "j:" + j);

i += 2;//i = i + 2

j += i;//j = j + i

}

public static int test1()

{

System.out.println("test1:" + "i:" + i + "," + "j:" + j);

i += 3;

j += i;

main(null);

System.out.println("test1Again:" + "i:" + i + "," + "j:" + j);

i += 4;

j += i;

return i + j + 5;

}

public static int test2()

{

System.out.println("test2:" + "i:" + i + "," + "j:" + j);

i += 6;

j += i;

main(null);

System.out.println("test2Again:" + "i:" + i + "," + "j:" + j);

i += 7;

j += i;

return i + j + 8;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

i += 9;

j += i;

//System.out.println("main end:" + "i:" + i + "," + "j:" + j);

}

}

**Example 1**

class A

{

static int i;

public static void main(String[] args)

{

System.out.println(i);

}

}

**Example 2**

class B

{

static int m;

static double n;

static boolean o;

static char p;

public static void main(String[] args)

{

System.out.println(m);

System.out.println(n);

System.out.println(o);

System.out.println(p);

}

}

**Example 3**

class C

{

static int i;

static int j;

static int k;

public static void main(String[] args)

{

System.out.println(i + "," + j + "," + k);

}

}

**Example 4**

class D

{

static int i,j,k;

public static void main(String[] args)

{

System.out.println(i + "," + j + "," + k);

}

}

**Example 5**

class E

{

static int i = 10;

static double j = 1.5;

public static void main(String[] args)

{

System.out.println(i + "," + j);

}

}

**Example 6**

class F

{

static int i = 10;

static double j = 1.5;

public static void main(String[] args)

{

System.out.println(i + "," + j);

i = 20;

j = 5.5;

System.out.println(i + "," + j);

}

}

**Example 7**

class G

{

static int i;

static double j;

public static void main(String[] args)

{

System.out.println(i + "," + j);

i = 20;

j = 5.5;

System.out.println(i + "," + j);

}

}

**Example 8**

class H

{

public static void main(String[] args)

{

int i = 10;

System.out.println(i);

i = 20;

System.out.println(i);

}

}

**Example 9**

class I

{

static int x = 20;

static double x = 5.5;

public static void main(String[] args)

{

System.out.println(x);

}

}

**Example 10**

class J

{

static int x = 20;

static double x = 5.5;

public static void main(String[] args)

{

System.out.println("done");

}

}

**Example 11**

class K

{

static int x = 20;

public static void main(String[] args)

{

System.out.println(x);

int x = 30;

System.out.println(x);

}

}

**Example 12**

class L

{

static int x = 20;

public static void main(String[] args)

{

int x = 30;

System.out.println(x);

System.out.println(x);

}

}

**Example 13**

class M

{

static int x = 20;

public static void main(String[] args)

{

int x = 30;

System.out.println(x);

System.out.println(M.x);

}

}

**Example 14**

class N

{

static int x = 20;

public static void main(String[] args)

{

int x = 30;

System.out.println(N.x);

System.out.println(x);

N.x = 200;

x = 300;

System.out.println(N.x);

System.out.println(x);

}

}

**Example 15**

class O

{

static int x = 20;

public static void main(String[] args)

{

System.out.println(O.x);

O.x = 200;

System.out.println(O.x);

}

}

**Example 16**

class P

{

public static void main(String[] args)

{

static int x = 20;

System.out.println(x);

}

}

**Example 17**

class Q

{

static int x;

x = 20;

public static void main(String[] args)

{

System.out.println(x);

}

}

**Example 18**

class R

{

static int x = 20;

System.out.println(x);

public static void main(String[] args)

{

System.out.println(x);

}

}

**Example 19**

class S

{

static int i = 10;

static void test()

{

i = 20;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

test();

System.out.println("main2:" + i);

}

}

**Example 20**

class T

{

static int i = 10;

static void test()

{

int i = 20;

}

public static void main(String[] args)

{

System.out.println("main1:" + i);

test();

System.out.println("main2:" + i);

}

}

**Example 21**

class U

{

static int i = 10;

static void test()

{

int i = 20;

i = 40;

U.i = 200;

}

public static void main(String[] args)

{

System.out.println("main1:" + i);

test();

System.out.println("main2:" + i);

}

}

**Example 22**

class V

{

static int i = 10;

static void test()

{

System.out.println("test1:" + i);

int i = 20;

i = 40;

V.i = 200;

System.out.println("test2:" + i);

System.out.println("test3:" + V.i);

}

public static void main(String[] args)

{

int i = 5;

System.out.println("main1:" + i);

System.out.println("main2:" + V.i);

test();

System.out.println("main3:" + i);

System.out.println("main4:" + V.i);

}

}

**Example 23**

class Char

{

public static void main(String[] args)

{

System.out.println("a\u0009b");

}

}

**Example 24**

class A

{

static void test()

{

System.out.println(i);

}

static int i = 10;

public static void main(String[] args)

{

System.out.println(i);

test();

System.out.println(i);

}

}

**Example 25**

class B

{

static void test()

{

System.out.println(i);

i = 100;

}

static int i = 10;

public static void main(String[] args)

{

System.out.println(i);

test();

System.out.println(i);

}

}

**Example 26**

class C

{

static int i = test();

public static int test()

{

return 20;

}

public static void main(String[] args)

{

System.out.println(i);

}

}

**Example 27**

class D

{

public static int test()

{

return 20;

}

static int i = test();

public static void main(String[] args)

{

System.out.println(i);

}

}

**Example 28**

class E

{

static int i = 10;

static int j = i;

public static void main(String[] args)

{

System.out.println(i);

System.out.println(j);

}

}

**Example 29**

class F

{

static int j = i;

static int i = 10;

public static void main(String[] args)

{

System.out.println(i);

System.out.println(j);

}

}

**Example 30**

class G

{

static int i=10;

static int j;

static int k = j;

static int m = i;

static int n= i+j+k+m;

public static void main(String[] args)

{

System.out.println(i);

System.out.println(j);

System.out.println(k);

System.out.println(m);

System.out.println(n);

}

}

**Example 31**

class H

{

static int i;

static int j = i + k;

static int k;

public static void main(String[] args)

{

System.out.println("done");

}

}

**Example 32**

class I

{

static int x = 10;

static int y = test();

static int test()

{

return 20;

}

public static void main(String[] args)

{

System.out.println(x);

System.out.println(y);

}

}

**Example 33**

class J

{

static int x = 10;

static int y = test();

static int test()

{

return x;

}

public static void main(String[] args)

{

System.out.println(x);

System.out.println(y);

}

}

**Example 34**

class K

{

static int x = test();

static int y = 10;

static int test()

{

return y;

}

public static void main(String[] args)

{

System.out.println(x);

System.out.println(y);

}

}

**Example 35**

class L

{

static int x = test();

static int y = 10;

static int test()

{

System.out.println("test: " + x + " , " + y);

return y;

}

public static void main(String[] args)

{

System.out.println(x);

System.out.println(y);

}

}

**Example 36**

class M

{

static int x = test();

static int y = test();

static int test()

{

System.out.println("test: "+ x + " , " + y);

return x + y + 10;

}

public static void main(String[] args)

{

System.out.println(x);

System.out.println(y);

}

}

**Example 38**

class N

{

static

{

System.out.println("N.SIB");

}

public static void main(String[] args)

{

System.out.println("N.main");

}

}

**Example 39**

class O

{

static

{

System.out.println("O.SIB1");

}

public static void main(String[] args)

{

System.out.println("O-main");

}

static

{

System.out.println("O.SIB2");

}

}

**Example 40**

class P

{

static int i = test();

static

{

System.out.println("P.SIB");

}

public static int test()

{

System.out.println("test");

return 69;

}

public static void main(String[] args)

{

System.out.println("P.main:" + i);

}

}

**Example 41**

class Q

{

static int i = test();

static

{

System.out.println("SIB:" + i);

i = 10;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

}

public static int test()

{

System.out.println("test:" + i);

return 69;

}

}

**Example 42**

class R

{

static int i = test();

static

{

System.out.println("SIB:" + i);

i = 10;

}

public static int test()

{

System.out.println("test:" + i);

return 69;

}

public static void main(String[] args)

{

System.out.println("main begin:" + i);

i = 33;

System.out.println(test());

System.out.println("main end:" + i);

}

}

**Example 43**

class S

{

static int i = test();

public static int test()

{

System.out.println("test:" + i);

i = 99;

main(null);

return 10;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

i = 33;

}

}

**Example 44**

class T

{

static int i = test();

static

{

System.out.println("SIB:" + i);

i = 99;

main(null);

System.out.println("SIBagain:" + i);

i = 10;

}

public static int test()

{

System.out.println("test:" + i);

i = 43;

main(null);

System.out.println("testagain:" + i);

return 75;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

i = 33;

}

}

**Example 45**

class U

{

static int i = test1();

static int j = test2();

static

{

System.out.println("sib:" + "i:" + i + ", " + "j:" + j);

i += 1;

j += i;

main(null);

System.out.println("sibAgain:" + "i:" + i + "," + "j:" + j);

i += 2;

j += i;

}

public static int test1()

{

System.out.println("test1:" + "i:" + i + "," + "j:" + j);

i += 3;

j += i;

main(null);

System.out.println("test1Again:" + "i:" + i + "," + "j:" + j);

i += 4;

j += i;

return i + j + 5;

}

public static int test2()

{

System.out.println("test2:" + "i:" + i + "," + "j:" + j);

i += 6;

j += i;

main(null);

System.out.println("test2Again:" + "i:" + i + "," + "j:" + j);

i += 7;

j += i;

return i + j + 8;

}

public static void main(String[] args)

{

System.out.println("main:" + i);

i += 9;

j += i;

}

}

**Example 46**

class V

{

public static void main(String[] args)

{

System.out.println("V.main");

}

}

class W

{

public static void main(String[] args)

{

System.out.println("W.main");

}

}

**Example 47**

class A

{

static int i;

static

{

System.out.println("A.SIB");

}

}

class V

{

static

{

System.out.println("V.SIB");

}

public static void main(String[] args)

{

System.out.println("main begin");

System.out.println(A.i);

System.out.println("main end");

}

}

**Example 48**

class A

{

static int i = 10;

static

{

System.out.println("A.SIB:" + i);

i = 20;

}

}

class W

{

static

{

System.out.println("W.SIB");

}

public static void main(String[] args)

{

System.out.println("main begin");

System.out.println(A.i);

System.out.println("---------------");

System.out.println(A.i);

System.out.println("---------------");

System.out.println("main end");

}

}

**Example 49**

class A

{

static int i;

static

{

int i = 10;

System.out.println("A.SIB:" + i);

System.out.println("A.SIB:" + A.i);

i = 20;

A.i = 200;

}

}

class X

{

static

{

System.out.println("X.SIB");

}

public static void main(String[] args)

{

System.out.println("main begin");

System.out.println(A.i);

A.i = 300;

System.out.println("---------------");

System.out.println(A.i);

A.i += 300;

System.out.println("---------------");

System.out.println(A.i);

System.out.println("main end");

}

}

**Example 50**

class A

{

static

{

System.out.println("A.SIB");

}

static void test()

{

System.out.println("A.test()");

}

}

class Z

{

static

{

System.out.println("Z.SIB");

}

public static void main(String[] args)

{

System.out.println("Z.main begin");

A.test();

System.out.println("----------");

A.test();

System.out.println("----------");

A.test();

System.out.println("Z.main end");

}

}

**Example 51**

class A

{

static int i;

static

{

System.out.println("A.SIB");

}

static void test()

{

System.out.println("A.test()");

}

}

class Z1

{

static

{

System.out.println("Z1.SIB");

}

public static void main(String[] args)

{

System.out.println("Z1.main begin");

A.test();

System.out.println("----------");

System.out.println(A.i);

System.out.println("----------");

A.test();

System.out.println(A.i);

System.out.println("----------");

A.test();

System.out.println("Z.main end");

}

}

**Example 52 (NEW PACK)**

class A

{

public static void test()

{

System.out.println("A.test()");

}

}

**Example 53**

class B

{

public static void main(String[] args)

{

System.out.println("B.main begin");

A.test();

System.out.println("B.main end");

}

}

**Example 54 (Saving as Test.java)**

class C

{

static int i = 20;

}

**Example 55**

class D

{

public static void main(String[] args)

{

System.out.println(C.i);

A.test();

}

}

class P

{

public static void main(String[] args)

{

System.out.println("main begin");

test();

System.out.println("main end");

}

public static void test()

{

System.out.println("test begin");

if(true)

{

System.out.println("if block");

return;

}

System.out.println("test end");

}

}

class Z3

{

public static void main(String[] args)

{

System.out.println("main begin");

int i = 10 + test();

System.out.println("----------------");

System.out.println(test() + " in the main@a");

System.out.println("----------------");

System.out.println(i + test() + " in the main@b");

System.out.println("----------------");

System.out.println(i + test() + test() + " in the main@c");

System.out.println("----------------");

System.out.println("@d:" + i + test() + test() + test());

System.out.println("main end");

}

public static int test()

{

System.out.println("from test");

return 100;

}

}

class Z42

{

public static void main(String[] args)

{

System.out.println("main begin");

int i = 1;

test(--i);

System.out.println("main end:" + i);

}

public static void test(int i)

{

System.out.println("from test:" + i);

++i;

}

}

class Z50

{

public static void main(String[] args)

{

System.out.println("main begin");

int i = 1;

int j = test1(++i) + i + test2(i++) + i + test3(i++) + i + test4(++i) + i +

test1(i++) + i + test2(++i) + i + test3(++i) + i + test4(i++) + i;

System.out.println("main end:" + i + "," + j);

}

public static int test1(int i)

{

return ++i;

}

public static int test2(int i)

{

return i++;

}

public static int test3(int i)

{

return i--;

}

public static int test4(int i)

{

return --i;

}

}

**23.04.21 Exercises**

1. Given 2 ints, a and b, return true if one if them is 10 or if their sum is 10.

makes10(9, 10) → true

makes10(9, 9) → false

makes10(1, 9) → true

2. We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return true if we are in trouble.

parrotTrouble(true, 6) → true

parrotTrouble(true, 7) → false

parrotTrouble(false, 6) → false

3. Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

diff21(19) → 2

diff21(10) → 11

diff21(21) → 0

4. Given two int values, return their sum. Unless the two values are the same, then return double their sum.

sumDouble(1, 2) → 3

sumDouble(3, 2) → 5

sumDouble(2, 2) → 8

5. We have two monkeys, a and b, and the parameters aSmile and bSmile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return true if we are in trouble.

monkeyTrouble(true, true) → true

monkeyTrouble(false, false) → true

monkeyTrouble(true, false) → false

6. The parameter weekday is true if it is a weekday, and the parameter vacation is true if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in.

sleepIn(false, false) → true

sleepIn(true, false) → false

sleepIn(false, true) → true