Operators :

Arithmetic(+,-,/,\*,%)

Logical

Incrementation and Decre-mentation

Assignment

Conditional

Bitwise Shift

Relational

Arithmetic Operators:

A screenshot of a computer program

Description automatically generated

The ones on which we are performing Operation we call them as Operands.

Logical : (&&,||,!)

Relational ( > ,<,==,>=,<=,!=)

Chained Assignment:

int a,b,c,d;

a=b=c=d=10;// This we will call it as Chained Assignment

All the variables a,b,c,d will hold 10;

A screenshot of a computer program

Description automatically generated

Compound Assignment :

A screen shot of a computer code

Description automatically generated

Unary: only one operand is sufficient to perform operation then we call it as unary

Assignment and incrementation & Decrementation ----Unary Operators

Conditional:

Only one block will get executed at a time.

int a=10;

int b=5;

int res;

if(a<b){

res=a-b;

}

else

{

res=a+b;

}

System.out.println(“res =”+res);////////15

A computer screen with text and numbers

Description automatically generated

Ternary Operator :(a>b)?a:b;

Switch Case:

A screenshot of a computer program

Description automatically generated

If the case value matches all the cases below it will also get executed.

Default is optional. This will also get executed.

If none of the cases matches default will get executed.

If we want to come out of switch after one of the case matches then we can go with switch.

A screenshot of a computer program

Description automatically generated

Default case can be written anywhere in the Switch. It will be similar to other cases but doesn’t contain any value.

Default can have break in its case:

A screenshot of a computer program

Description automatically generated

In order to come out after a case match then we need to keep break.

A screenshot of a computer program

Description automatically generated

We cannot have duplicate case it will result in compile Time Error.

A screenshot of a computer program

Description automatically generated

For all the inputs coming as Command Line Arguments memory will get created.

But we cannot assign a value to unallocated memory variable.

In the above case for Good --- it will be stored in args[0] as memory will be present.

There will be no memory present for the args[1] due to which JVM will throw Exception.

If JVM throws Error then we can call it as an ExceptionA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

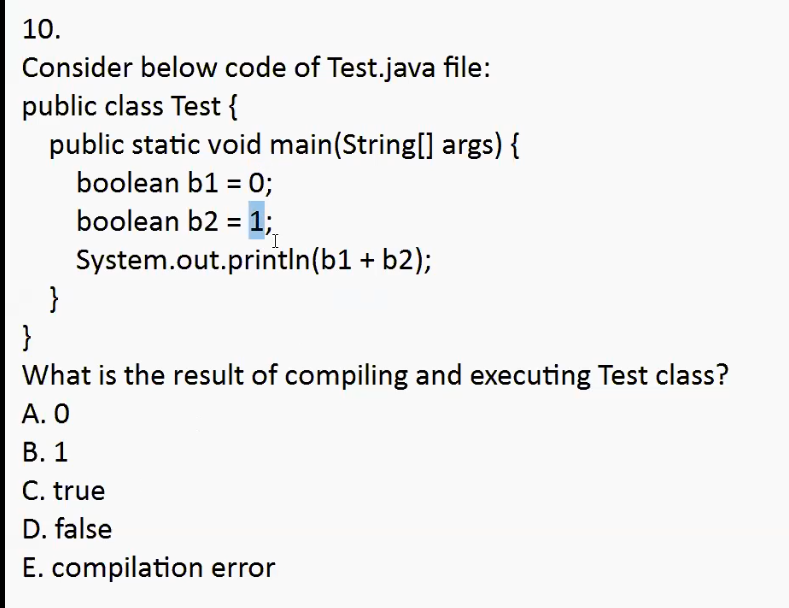
Description automatically generated

A black and white image of a square and a black rectangle

Description automatically generated

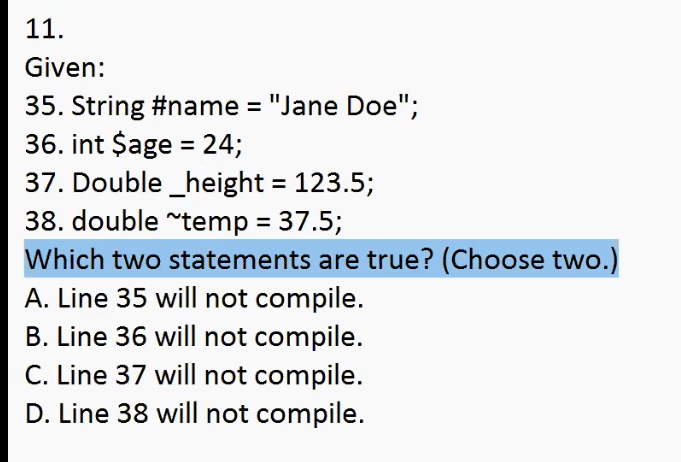
In Command Line Arguments if we want give data arguments that have spaces in it as a Single argument then we need to enclose them in Quotations

Then enter Data will be Considered as String.

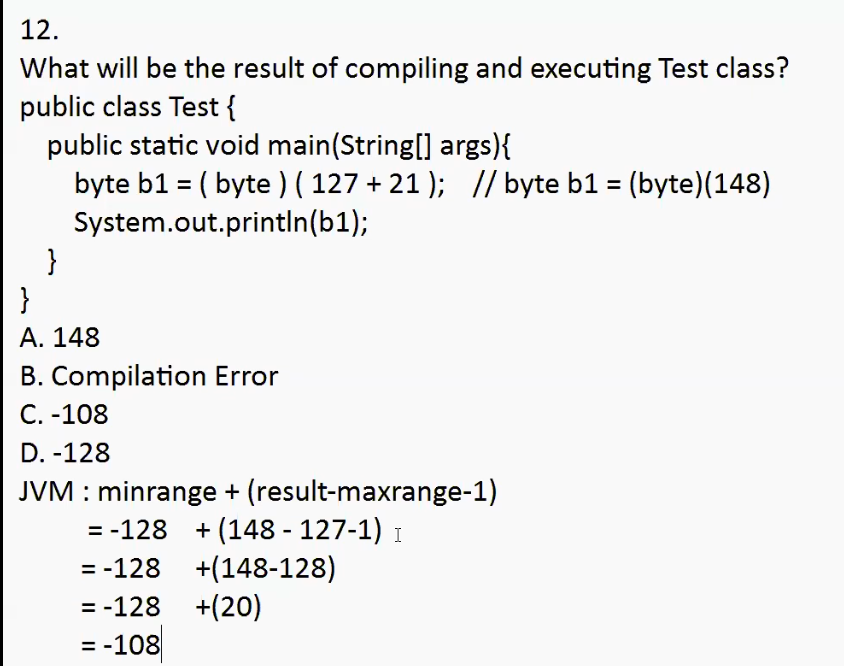


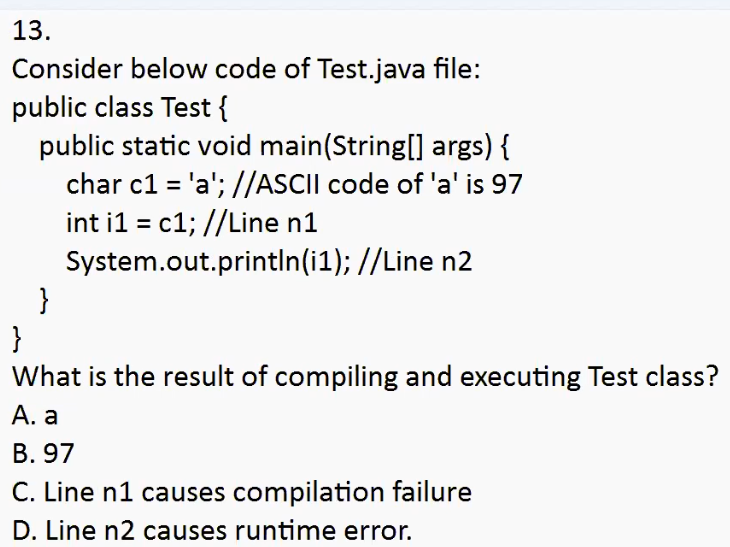
Compilation Error : In-Compatible Type

For Boolean data type we can have only true/false



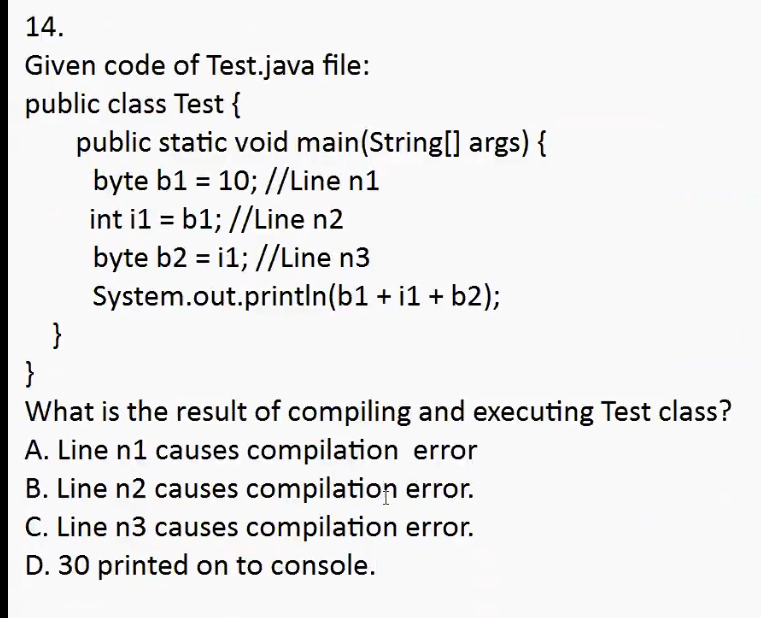
Ans: A,D

. 



charàint (implicit type casting)

Ans : B



Ans : C (Line n3 causes compilation error)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Type mismatch: cannot convert from int to byte

//If the value is literal then compiler will check if that value can be stored.

//If it is variable then compiler will check the data type of that variable.

A screenshot of a computer test

Description automatically generated

A=100

B=102

C=102

X=103

Ans:100

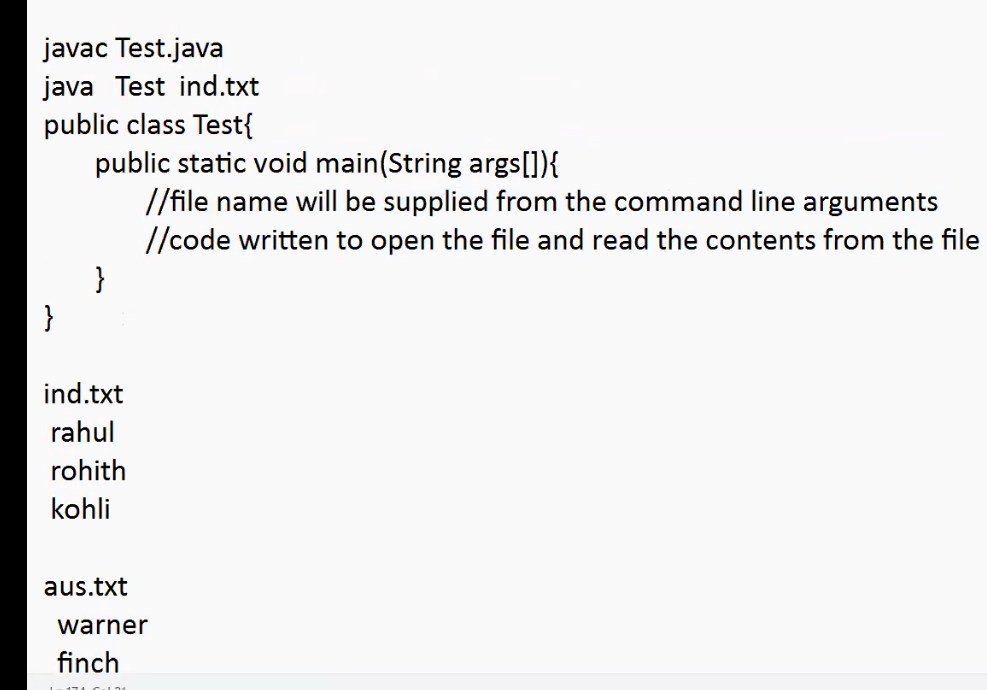
A screenshot of a computer code

Description automatically generated

Ans : -100

Compiler will do type checking

JVM will allocate the memory and few other things.



Code will take inputs from file and read the contents.

Command Line Arguments can be used to read the contents file.

A close up of a computer code

Description automatically generated

50+52—102

**public** **class** Switch {

**public** **static** **void** main(String args[])

{

**char** a='2';

System.***out***.println(a++ + ++a);

}

}

Initially value of a will used for addition ‘2’

Then it will get incremented to ‘3’

Then we have ++a due to which the addition will be like ‘2’ + ‘4’

50 + 52 --102