```
Q>
Consider below code:
public class Test {
    public static void main(String[] args) {
        char c = 'Z';
        long l = 100_00l;//from JDK1.7 for a literal we can give _ also, if we give
compiler will remove that _ in .class file
        int i = 9_2;//from JDK1.7 for a literal we can give _ also, if we give
compiler will remove that _ in .class file
        float f = 2.02f;
        double d = 10_0.35d;//from JDK1.7 for a literal we can give _ also, if we
give compiler will remove that _ in .class file

        l = c + i;//char + int = int   int -----> long (implicit)
        f = c * l * i * f;//char * long*int*float = float
        f = l + i + c;//long+int+char = long  long---> float(implicit)
        i = (int)d;//double----> int(explicit)
        f = (long)d;//double---> long  , long ---> float (implicit)
    }
}
Does above code compile successfully?
A. Yes
B. No
Answer : A


Q>
class Test
{
      public static void main(String[] args)
      {
            int a = 20;   // a = 18
            int var= --a * a++ + a-- - --a;//  var = 19 * 19 + 20 -18 = 363
            System.out.println("a = " + a);// a = 18
            System.out.println("var = " + var);//var= 363
      }
}
A.
 a = 18
 var=363
B.
    a = 363
    var=363
C. Compilation Error
D.
  a = 25
var= 363

answer : A

Q>
class Test
{
      public static void main(String[] args)
      {
            int i = 5; // i  = 5,6,7
               //5 < 6(true)
            if (i++ < 6)
            {
                  System.out.println(i++);//System.out.println(6)
            }
      }
}
```
A. 5
B. 6
C. Program executes succesfully but nothing is printed on to console
D. 7

Q>
```
int x  = 4;//line-n1
int y = 4++;//line-n2   whether it is post or pre-increment it can only be done on variables not on direct literals
System.out.println(x);
System.out.println(y);
```

A. line-n1 CompileTimeError
B. x=4
     y=5
C. x=5
     y=5
D. line-n2 CompileTimeError

Answer: D

Q>
24
```
 int x = 4;//line-n1
 int y =++(++x);//line-n2 whether it is post or pre-increment it can only be done on variables not on direct literals
 System.out.println(x);
 System.out.println(y);
```

A. line-n1 CompileTimeError
B. x=4
     y=5
C. x=5
     y=5
D. line-n2 CompileTimeError

Answer: D

Q>
```
boolean b=true;//line -n 1
 b++;//line-n2    Ans. increment and decrement is applicable only for integral,floating type and character type not for boolean type
 System.out.println(b);
```

A. line-n1 Compile Time Error
B. line-n2 Compile Time Error
C. false
D. true
E. None of the above

Anser : B

Q>
```
  int b,c,d;//declaring the variables
  int a= b = c = d=10;/intializing the values for b=c=d and finally giving the
value to a variable with declaration
  Will the code compile?
```

A. yes
B. no

Answer: A

Q>
```
  int a = b = c = d = 20;//b,c,d not declared but using so CompileTime Error
  System.out.println(a);
  Will the code compile?
```

A. yes
B. no

Answer : B

Q>
```
 byte c = (10> 20) ? 30 : 40;//literals are involved so compiler performs operation
                                    byte c =40;
 byte d =(10<20) ? 30 : 40;//literals are involved so compiler performs operation
                                    byte d =30;
 System.out.println(c);//40
 System.out.println(d);//30
```

```
A. 30
     40

B. 40
     30

C. 10
     20

D. 20
     10

E. CompiletimeError
```

Answer : B

Q>
```
 int a = 10, b = 20;//type checking is valid no problem
 byte c = (a>b) ? 30 : 40;//literals are not involved in operation, so compiler
would just check type checking of result
                                    Compiler will see 30,40 type it knows is int,
so the result should be of int type only.
                                    if compiler only perform the operation it will
try to map with casting chart otherwise it wants
                                   the exact type.
 byte d =(a<b)  ? 30  : 40;
 System.out.println(c);
 System.out.println(d);
```

```
A. 30
     40
```

```
B. 40
     30

C. 10
     20

D. 20
     10

E. CompiletimeError

Answer: E


Q>
Consider below statements:
1. int x = 5____0;// Literal values can be with _  also but it should be in b/w not
at the beginging or at the end
2. int y = ____50;//invalid becoz starts with _
3. int z = 50____;//invalid becoz in ends with _
4. float f = 123.76_86f;//valid
5. double d = 1_2_3_4;//valid

How many statements are legal?
A. One statement only
B. Two statement only
C. Three statement only
D. Four statement only
E. All 5 statement only.

Answer : C


Note:
Compiler -> it checks only the syntax and makes the jvm execution smoothful
JVM          -> Creates the memory for the variables and perform type casting and
generates the result.
```