Important tricks for MID by Riyadh 

LOGICAL ERROR : the program runs but provides wrong output.

Example :

```
int num = 2;

if ( (num % 2) == 0)
{
      System.out.println("the number is odd");
}
```

RUN-TIME ERROR : the program stops running suddenly when asking the OS executing a non accepted statement.

Example :

```
int num = 40;

System.out.println(num/0);

}
```

COMPILEATION ERROR : Grammatical mistakes in a program.

Examples :

```
System.out.prin("Hello world");

integer num = 90;

double num = 9.0
```

Concatenation of strings and integers :

```
System.out.println(4 + 5 + "Hello");
```

Output : 9Hello

```
System.out.println("Hello" + 5 + 4 );
```

Output : Hello54

```
System.out.println("Hello"+ (5+4) );
```

Output : Hello9

Show the result of the following code :

a)
```
System.out.println(2 * (5 / 2 + 5 / 2));

   // 2 * (2 + 2) = 2(4) = 8 (int trick)
```

b)
```
System.out.println(2 * 5 / 2 + 2 * 5 / 2);

   // 10/2 + 10/2 = 5 + 5 = 10 (int trick) (operator precedence)
```

c)
```
System.out.println(2 * (5 / 2));

   // 2 * (2) = 4 (int trick) (operator precedence)
```

d)
```
System.out.println(2 * 5 / 2);

   // 10/2 = 5 (int trick) (operator precedence)
```

e)
```
System.out.println( 5 / 2 * 2);

   // 2*2 = 4 (int trick) (operator precedence)
```

f)
```
System.out.println(2 * (5.0 / 2 + 5.0 / 2));

   // 2 * (2.5 + 2.5) = 2 * (5.0) = 10.0
```

Are the following statements correct? If so, show the output:

a)
```
System.out.println("25 / 4 is " + 25 / 4);

   // 25 / 4 is 6 (int trick) (operator precedence) (Concatenation)
```

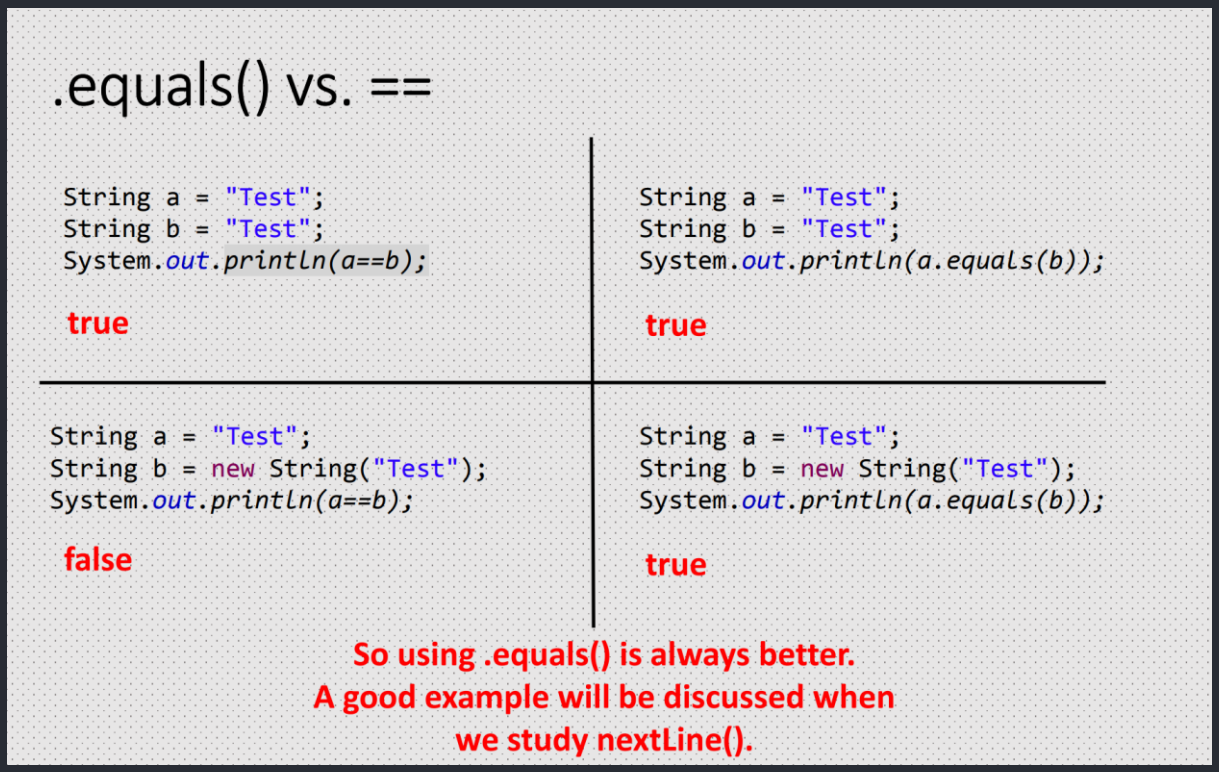
b)
```
System.out.println("25 / 4.0 is " + 25 / 4.0);

   // 25 / 4.0 is 6.25 (operator precedence) (Concatenation)
```

c)
```
System.out.println("3 + 2 is " + 3 + 2);

   // 3 + 2 is 32 (Concatenation)
```

d)
```
System.out.println("3 + 2 is " + (3 + 2));

   // 3 + 2 is 5 (Concatenation)
```

e)
```
System.out.println("3 * 2 / 4 is " + 3 * 2 / 4);

   // 3 * 2 / 4 is 1 (operator precedence) (Concatenation)
```

f)
```
System.out.println("3.0 * 2 / 4 is " + 3.0 * 2 / 4);

   //3.0 * 2 / 4 is 1.5
```



Increment & decrement operators :

```
      int count = 1;
      int num = ++count;
      // num = 2 , count = 2
is equivalent to :
      int count = 1;
      count += 1 ;
      int num = count;
      // num = 2 , count = 2
and

      int count = 1;
      int num = count++;
      // num = 1 , count = 2
is equivalent to :
      int count = 1;
      int num = count;
      count += 1 ;
```

// The same applies to the decrement operator (count–- and --count)

nextLine() problem :

```
System.out.println("enter num : ");
int num = input.nextInt();

System.out.println("Enter str : ");
String str = input.nextLine();

System.out.println("num = " + num +" str =" + str);
```

Output :

```
enter num :
115
Enter str :
num = 115 str =
```

if a method of a Scanner came before the nextLine() , like nextInt() , next() , nextDouble() then it will skip the method nextLine()

the solution :

```
System.out.println("enter num : ");
int num = input.nextInt();

input.nextLine();

System.out.println("Enter str : ");
String str = input.nextLine();

System.out.println("num = " + num +" str =" + str);
```

Output :

```
enter num :
115
Enter str :
Hello
num = 115 str =Hello
```

Here the Scanner will skip the first nextLine(), therefore the other nextLine() will not affect by nextInt()

Syntax of The do-while statement :

```
do
     Body_Statement

while (Boolean_Expression);
```

Don’t forget the semicolon!

while vs. do-while

```
public class WritewhileAnddowhileLoops {
    public static void main (String[] args) {
        int i=0;
        System.out.println("Try while loop:");
        while (i < 5) {
            System.out.println("Iteration " + ++i);
        }
        System.out.println("Try do while loop:");
        i=0;
        do {
            System.out.println("Iteration " + ++i);
        }
        while (i < 5) ;
    }
}
```

```
Try while loop:
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Try do while loop:
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
```

```
public class WritewhileAnddowhileLoops {
    public static void main (String[] args) {
        int i=5;
        System.out.println("Try while loop:");
        while (i < 5) {
            System.out.println("Iteration " + ++i);
        }
        System.out.println("Try do while loop:");
        i=5;
        do {
            System.out.println("Iteration " + ++i);
        }
        while (i < 5) ;
    }
}
```

```
Try while loop:
Try do while loop:
Iteration 6
```

25

for statement local variable :

```
for (int n = 1 ; n <= 10 ; n++)
      System.out.println("hello");

n = 9;
```

```
// here we will get an error because n is local to the loop
```

The exit method :

The exit Method

- Sometimes a situation arises that makes continuing the program pointless.
- A program can be terminated normally by `System.exit(0)`.

36

```
int a =4;

if (a>=4) {
      System.out.println("Hello ! ");
      System.exit(0);
}
else
      System.out.println("greetings!");
System.out.println("print me ? ");
```

Short-circuit evaluation:

Short-circuit evaluation

- Short-circuit evaluation is not only efficient, sometimes it is essential!
- A run-time error can result, for example, from an attempt to divide by zero.

```
if ((number != 0) && (sum/number > 5))
```

- *Complete evaluation* can be achieved by substituting `&` for `&&` or `|` for `||`.

44

10. What is the output of this program?

```
class Output    {
   public static void main(String args[])       {
      int x , y = 1;
      x = 10;
      if (x != 10 && x / 0 == 0)
         System.out.println(y);
      else
         System.out.println(++y);
   }    }
```

a) 1
b) Runtime error owing to division by zero in if condition
c) 2
d) Syntax error

The switch statement :

Syntax :

```
switch(Controlling_Expression)
{
case Case_Label:
Statement(s);
break;
case Case_Label:
…
default:
…
}

// The default case is optional, but recommended, even if
it simply prints a message.

// ALSO THE SWITCH STATEMNT DOES NOT ACCEPT A FOLATING-POINT
VARIABLE

// The optional break statement prevents the
consideration of other cases.
```

9. What is the output of the following code segment?

```
public class MyClass {
   public static void main(String [] args) {
    char x = 'A';
while(x != 'D'){
        switch(x){
                case 'A':
                        System.out.print(x);
                        x = 'D';
                case 'B':
                        System.out.print(x);
                        x = 'C';
                        break;
                case 'C':
                        System.out.print(x);
                        x = 'D';
                default:
                        break;
}}}
  }
```

Scope trick :

2. What is the output of this program?

```
class variable_scope{
    public static void main(String args[])        {
        int x = 5;
        {
            int y = 6;
            System.out.print(x + " " + y);
        }
        System.out.println(x + " " + y);
    }    }
```

a) 5 6 5 6

b) 5 6 5

c) Runtime error

d) Compilation error

bracket problem

// here y is local variable inside the bracket, we cannot call it outside the brackets, so it is (D)

17) What will be the output of the following code?

```
for(int i=1; i<=10; i++);
System.out.print(i);
```

a) 12345678910

b) 11

c) Error

d) 1 2 3 4 5 6 7 8 9 10

Idention trick :

12. What output is produced by the segment of code shown below:

```
int x = 12;
if (x > 12)
if ( x < 15)
System.out.print("BLUE");
else
System.out.print("GREEN");
System.out.print("JEANS");
```

a) BLUE b) GREENJEANS

c) JEANS d) BLUEJEANS

16) Which of the following three `if` statements are equivalent?

```
1. if (a == b){
            if (c == d)
                a = 1;
            else b = 1;}
2. if (a == b)
            if (c == d) a = 1;
else b = 1;
3. if (a == b){
            if (c == d)
                a = 1;}
            else b = 1;
```

a) 1 and 2
b) 1 and 3
c) 2 and 3
d) None of them are equivalent

Using String method by char variables trick :

3) What is the output of the following program?

```
public class CSC111Mid1Questions {

    public static void main(String[] args)
    {
        String _aA = "123";
        char $1 = '1';

        System.out.println(_aA.charAt(0).equals($1);

    }

}
```

a) Runtime error
b) Compilation error
c) 1
d) 2

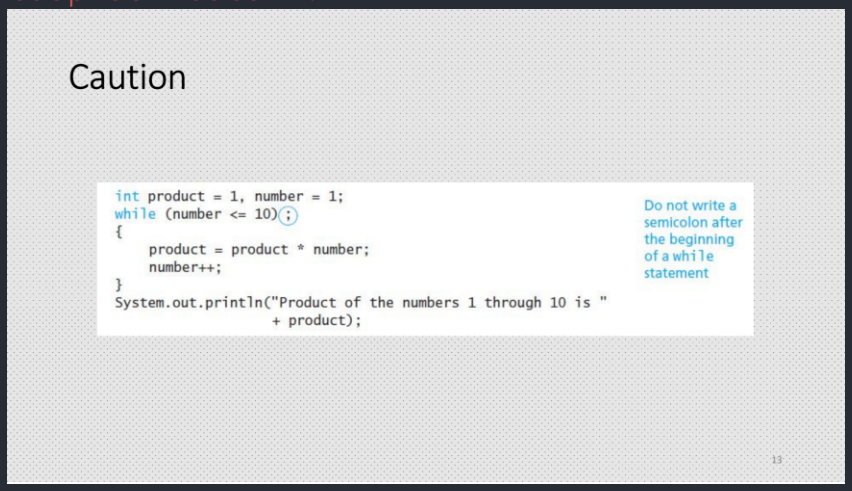
Reading from right to left trick :

5) What is the output of the following program?

```
public class CSC111Mid1Questions {

    public static void main(String[] args)
    {
     int a,b,c,d;
        a=b=c=d=20;
        a+=b-=c*=d%=20;
        System.out.println(a+" "+b+" "+c+" "+d);

    }

}
```

a) 40 20 0 0
b) Compilation error
c) Runtime error
d) 20 0 0 20

loop semicolon :




```
int a = 4;
while(a-- > 1);
{
    System.out.println(a);
}
```

18) What will be the output of the following code?

```
int i;
for(i=1; i<=10; i++);

System.out.print(i);
```

a) 12345678910
b) 11
c) Error
d) 1 2 3 4 5 6 7 8 9 10

Nested loop easy solution :

- How many times will the string “Here” be printed?

```
int count1 = 1, count2 = 1;
while(count1 <= 10){
        count2 = 1;
        while(count2 <= 20){
                System.out.println("Here");
                count2++;
        }
        count1++;
}
```

10 * 20 = 200

Boolean ! :

9) What is the output of the following program:

```
boolean var1 = false;
boolean var2 = true;
if (!var1)
        System.out.println(var1);
else
        System.out.println(var2);
```

a) 0
b) 1
c) true
d) false