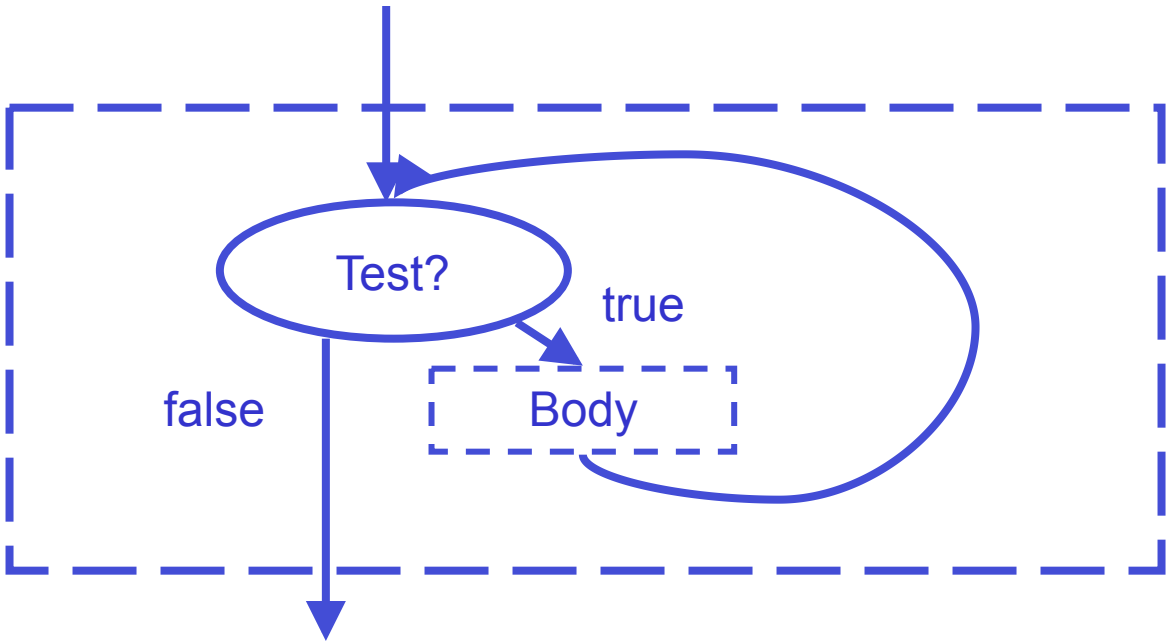
ITI 1120
Lab #5

Loops in Java

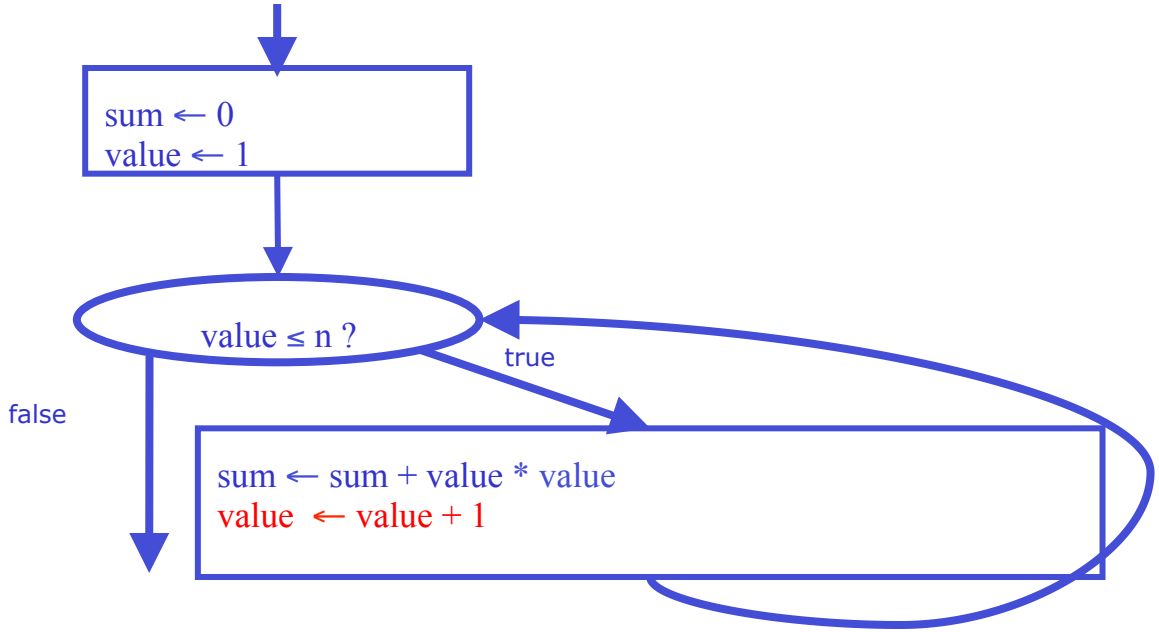


- Java

```
while (Test)
{
    Body
}
```

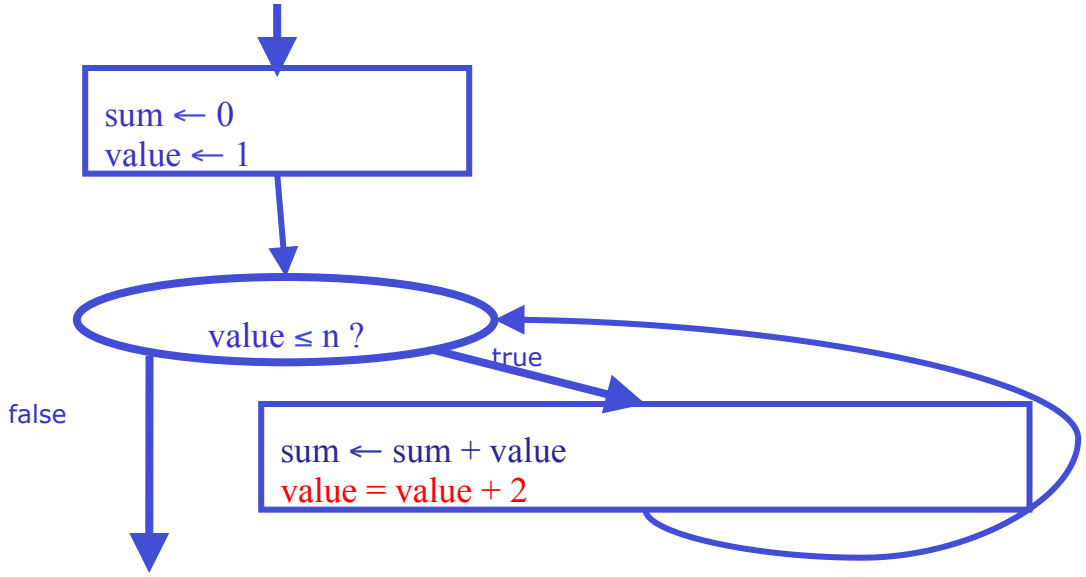
Sum of Squares

GIVENS:	n	(a number ≥ 1)
RESULTS:	sum	(sum of squares from 1^2 to n^2)



Sum of Odd Integers

GIVENS:	n	(a number ≥ 1)
RESULTS:	sum	(sum of odd integers up to n)



Sum of odd integers

What does this program print for n =5. Let's execute this program on paper.

```
1  public class  odd{
2      public static void main(String[] args) {
3          System.out.println("Enter a positive interger");
4          int n = ITI1120.readInt();
5          int sum, value;
6
7          sum=0;
8          value=1;
9          while(value<=n)
10         {
11             sum=sum+value;
12             value=value+2;
13         }
14
15         System.out.println("The sum of all odd positive"+
16             " integers in the range [1,"+n+"] is "+sum);
17     }
18 }
```

Trace for Sum of Odd Integers

Statement	n	value	sum
(User entered 5) Initial values	5	?	?
Line 7: sum = 0			0
Line 8: value = 1		1	
Line 9: value <=n ? (1 <= 5) true			
Line 11: sum = sum + value			1
Line 12: value = value + 2		3	
Line 9: value <=n ? (3 <= 5) true			
Line 11: sum = sum + value			4
Line 12: value = value + 2		5	
Line 9: value <=n ? (5 <= 5) true			
Line 11: sum = sum + value			9
Line 12: value = value + 2		7	
Line 9: value <=n ? (5 <= 5) false			
Line 15: System.out.println(…)			

Exercise 1 (on paper)

```
1  import java.util.Scanner;
2  public class lab5_1 {
3    public static void main(String[] args) {
4      Scanner keyboard=new Scanner(System.in);
5
6      int number, apple;
7      number=keyboard.nextInt();
8      apple=number;
9
10     while(number !=0){
11       number=keyboard.nextInt();
12       if(number > apple){
13         apple=number;
14       }
15     }
16     System.out.println("apple is " + apple);
17   }
18 }
```

- What does the program on the left print if the input user entered was:
  1 5 -3 0
- (make a table as in the previous example, if uncertain)
- Can you tell what the program does? Write one sentence explaining it in plane English.

Exercise 2 (on paper)

Analyze the following code. Is count<100 always true, always false or sometimes true and sometimes false at Point A? Answer the same question for Point B and Point C?

```
int count =0;
while(count <100){
   //Point A
   System.out.println("hello word");
   count++;
   //Point B
}
//Point C
```

Exercise 3 (on paper)

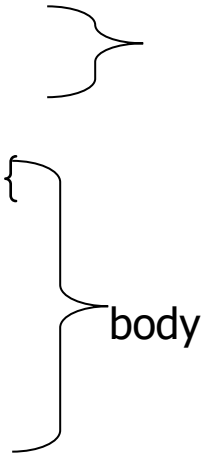
A programmer who wrote a method below thinks that the method tests if a number is a prime. However there is a logical mistake.

1. Where is the mistake?
2. For what number does the method give a wrong answer?
3. Fix the mistake.

```
public static boolean isPrime(int n)
{
  for (int divisor = 2; divisor <= n/2; divisor++){
    if (n % divisor == 0){ // If true, number is not prime
      return false;
    }
    else{
      return true;
    }
  }
  return true;
}
```

`for` loop syntax

```
for (<initialization>; <test>; <update>) {
    <statement>;
    <statement>;
    ...
    <statement>;
}
```



- Perform ***<initialization>*** once.
- Repeat the following:
  - Check if the ***<test>*** is true. If not skip the loop.
  - Execute the ***<statement>***s.
  - Perform the ***<update>***.

Exercise 4 (on paper)

How many starts does the following program print?

```
for(int i = -2; i <= 13; i++) {
    System.out.print("*");
    System.out.print("**");
}
```

A. 0 B. 15 C. 45 D. 48 E. 68

Exercise 5 (on paper): Nested Loops

What does the following program print?

```
1  public class test {
2    public static void main(String[] args) {
3      int i=5;
4      while(i>=1){
5        int num =1;
6        for(int j=1; j<=i; j++){
7          System.out.print(num + " | ");
8          num = num*2;
9        }
10       System.out.println();
11       i--;
12     }
13   }
14 }
```

Programming Exercise 1

- Design a Java program, by using do-while loop that asks the user to enter two integers. The numbers should be added and the sum displayed. Then the user is asked to enter 'Y' if she wishes to perform the operation again, and otherwise she enters any character other than 'Y'. The operation is repeated, as long as the user chooses 'Y'.

Programming Exercise 2

- Design a Java program that asks the user for two positive integers no greater than 15 (length and width of a rectangle). The program should then display a rectangle of '*' on the screen with the corresponding dimensions. For example, if the user inputs width=5 and length=3, then the program should print:

  *****

  *****

  *****

  Your program should have a method called printRectangle that takes as input width and length and draws the rectangle.

Programming Exercise 3: Experiment with Perfect Numbers

- A positive integer is called a perfect number if it is equal to the sum of **all** of its positive divisors, **excluding itself**. For example, 6 is perfect number since 6=1+2+3. The next is 28=1+2+4+7+12. There are four perfect numbers less than 10,000. Write a program that finds all these four numbers.

- Once you are done. Modify your program so that it looks for all perfect numbers smaller than 35 million. What do you notice? Assuming that you computer can do billion instructions in a sec, can you figure out how long, roughly, will it take your computer to find 5th perfect number (it is 33,550,336). Is the answer roughly: couple of minutes, couple of hours, couple of days, … weeks, months, years ?
- What if you wanted to wait until it prints 6th perfect number, which is 8,589,869,056?

Arrays: Prog. Exercise 4: Fibonacci Numbers

Write a java program that creates an array containing n values (n is 2 or greater) such that

A[0] = 1

A[1] = 1

A[i] = A[i-1] + A[i-2]

for all i in the range 1 < i < n.

- For example, the array of length 7 would be:

A = 1 1 2 3 5 8 13

- Once you are done, trace your program on paper to see what the values in the array will be for n=5.