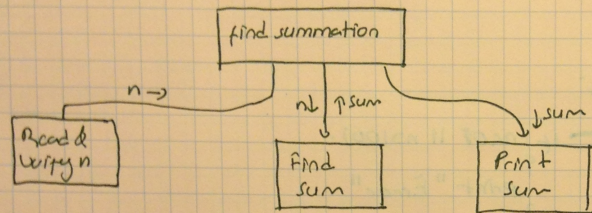


22/10/2015

code segment: not actual program
portable

→ sum: $\sum_{i=1}^n \sum_{j=1}^i j$ $0 < n \leq 100$

code segment



// Read user input

// Make sure that $0 < n \leq 100$

while method:

$\sum_{i=1}^{n=4} \sum_{j=1}^i j$

trace:

```

sum = 0;
i = 1;
while (i <= n)
{
    j = 1;
    while (j <= i)
    {
        sum = sum + j;
        j = j + 1;
    }
    i = i + 1;
}
  
```

n	sum	i	j
3	0	0	1
	1	2	2
	2	3	1
	4		2
	5		3
	7		1
	10		2
			3
			4

Another version:

```

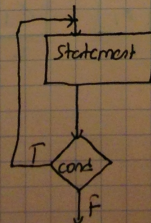
sum = 0;
i = 1;
while (i <= n)
{
    j = 1;
    while (j <= i)
    {
        if (sum < 5)
            sum = sum + j;
        j = j + 1;
    }
    i = i + 1;
}
  
```

→ We do testing at the end.

Do-Loop: (post-test)

```

do
{
    statement
}
while (cond);
  
```



→ Read & verify n.
 → Ask user the value of n.
 → Read n if it is correct

female } Robust program
 female }

come back!

Java

```
do
{
  System.out.println("Enter n");
  Read n
} while (n < 1 || n > 100);
n is valid;
```

if (n < 0 || n > 100).
 { print "Error"
 }

→ The problem with a different solution = this time we use while loop.

print "Enter a value"

Read n.

while (n < 0 || n > 100)

```
{
  print enter n value
  read n.
}
```

Print "your input is incorrect"

Problem with "Fibonacci Numbers"

0, 1, 1, 2, 3, 5, 8, ...

fib1 = 0
 fib2 = 1

i = 2

while (i < 10)

```
{
  fib3 = fib1 + fib2;
  print fib3;
  fib1 = fib2;
  fib2 = fib3;
  i = i + 1;
}
```

