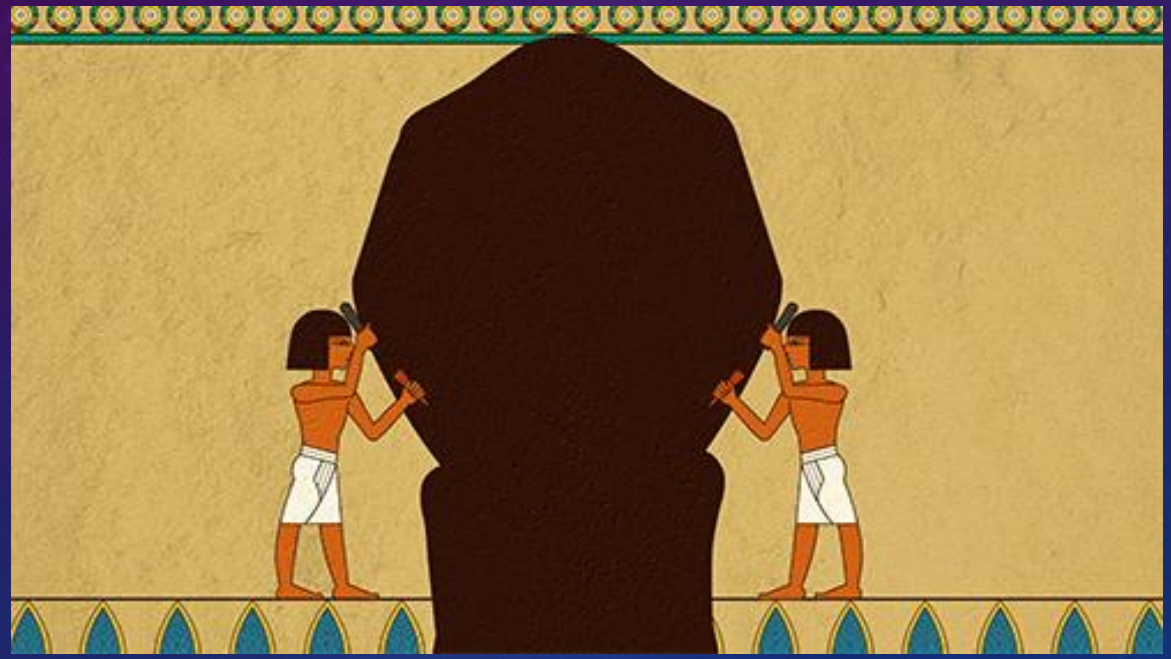
Get Smart: With Java Programming

Yaman Omar Alashqar

```
System.out.println("WELCOME TO THIS COURSE\n");
```

LECTURE



Random Numbers
Repetition Statements

--- For Loop ---

Branching statements

--- Break ---

--- Continue ---

Problem Solving

LOOPS

Counter Controlled Loop
Definite repetition

1) The user will enter the number of iterations (N) Entries

2) The number of iterations are predefined
   e.g. (50) Entries

3) The loop has a range from (x) to (y)
   e.g. Range from (5) to (30)

Condition Controlled Loop
Indefinite repetition

- When the number of iterations are unknown
- Loop obtains data in each iteration

- The user will enter a dummy value to end
  e.g. (Enter (-1) to end)
  e.g. (Enter ('e') to end)

Q) How can the program determine when to stop?
A) One way to solve this problem is to use a special (distinct) value called a sentinel value (also called a signal value, a dummy value, or a flag value) to indicate "end of data entry"

LOOPS

Counter Controlled Loop

Number of iterations are known before entering the loop

We will need

1) Initial value

2) A condition that tests for the final value

3) increment (or decrement) by which the control variable is modified each time through the loop (to make the condition false after many iterations)

Condition Controlled Loop

Number of iterations are unknown before entering the loop

We will need a dummy value to end the loop
(A distinct vale from other regular values)
The loop includes a statement that obtains data

OR we can use the break; statement
The loop includes a condition

➢ ‘break’ Statement

- Used inside the body of a loop statement or after a ‘case’ in a ‘switch’ statement
- Causes immediate exit from the loop or switch
- We use it when we want to exit early from the loop, by inserting it inside a condition statement
- Mostly used to exit after a ‘flag value’ is found

```
while (test Expression)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

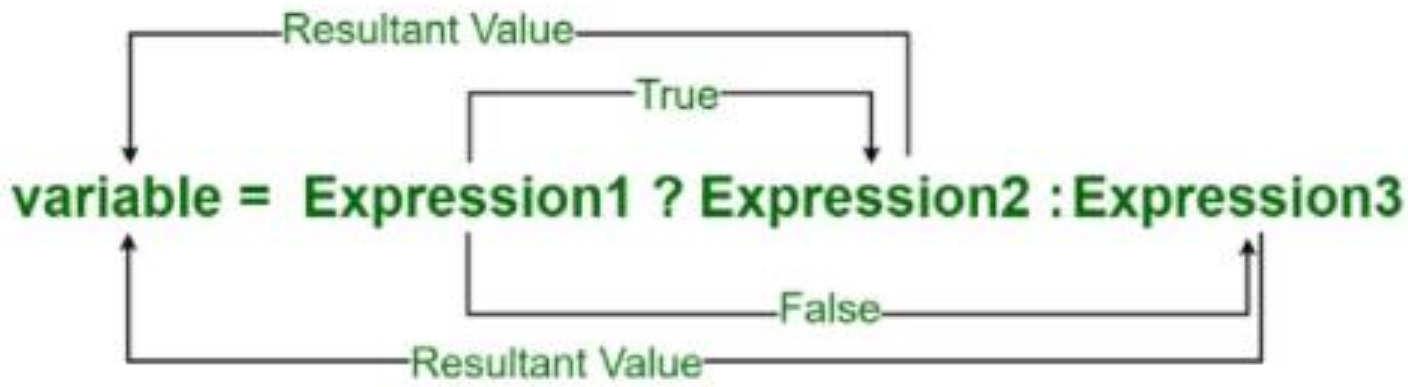
➢ ‘continue’ Statement

- Used inside the body of a loop statement
- Skips any statement after it within the loop and jumps back to the first line in the block of code in the loop
- Mostly used to skip the code in special cases, for example if we want to skip the code after a negative number [if(n<0) continue;]

```
while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

- Min / Max
- Number of digits
- Sum / AVG / Product

Ternary Operator



CHALLENGE:
FIND THE SECOND LARGEST NUMBER

{ MAX & SECOND MAX }

- Factorial
- Exponents
- Prime Numbers

ANSWER PERCENTAGE

- Number of people who answered “A”
- Number of people who answered “B”
- Number of people who answered “C”
- Number of people who answered “D”

Random Numbers

- Random Student Selector
- Random Password Generator

- Lottery (Lot applications)
- Video game (different events)

Random number generators have applications in statistical sampling, computer simulation, cryptography, completely randomized design

Random Numbers

First create an instance of “Random” class and then invoke methods such as nextInt(), nextDouble(), etc using that instance.

We can pass arguments to the methods for placing an upper bound on the range of the numbers to be generated. For example, nextInt(6) will generate numbers in the range 0 to 5 both inclusive.

```
// create instance of Random class
        Random rand = new Random();

        // Generate random integers in range 0 to --
        //(max - min) + 1
        int rand_int1 = rand.nextInt(20) ;
        int rand_int2 = rand.nextInt(10);

        // Print random integers
        System.out.println("Random Integers: "+rand_int1);
        System.out.println("Random Integers: "+rand_int2);

        // Generate Random doubles
        double rand_dub1 = rand.nextDouble();
        double rand_dub2 = rand.nextDouble();

        // Print random doubles
        System.out.println("Random Doubles: "+rand_dub1);
        System.out.println("Random Doubles: "+rand_dub2);
```

Random Student Selector

No Hands
Random Student Selector

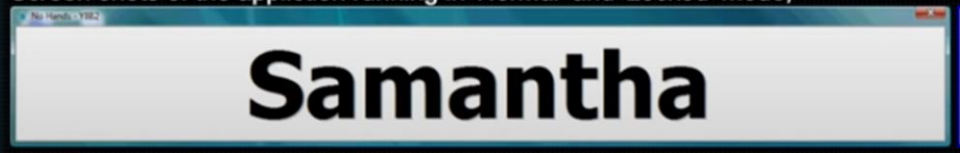
What Is 'No Hands'?

'*No Hands*' random student selector is a small windows utility that randomly selects a Name from a class list. It is based on the idea that by randomly selecting pupils to answer questions, students feel more comfortable in answering teachers questions, and *all* students are more attentive as anyone in the class may be asked the next question.

In practice, students love this quick and fair way of student selection, and teachers can never be accused of favouritism or picking on a student too frequently.

'No Hands' is particularly suited to use with a smart-board as the application can be loaded with many class lists (consisting of a list of names and times when the class is taught) positioned and locked on the smart-board. Then whenever you need to select a pupil, run the application and a student for the current class will be selected. Click again and another student will be picked. You can optionally lock the utility to the screen so that the little darlings don't close it when your back is turned, and you can make it sit on-top of your other windows. There is also a manual override for the automatic class selection.

Screen shots of the appliction running in '*Normal*' and '*Locked*' mode;



'No Hands' is a proven solution to a common classroom problem found in both primary and secondary schools. The utility already has an ever expanding user base, and I know from my own experience, that it is a great teaching tool to have in any classroom. Hope you like :)

Exercises

- Generate a random integer between 100 and 999 which is divisible by 5
- Generate a 6 digit random code
- Pick a random character from a given String
- Calculate multiplication of two random double numbers

MATH.RANDOM()

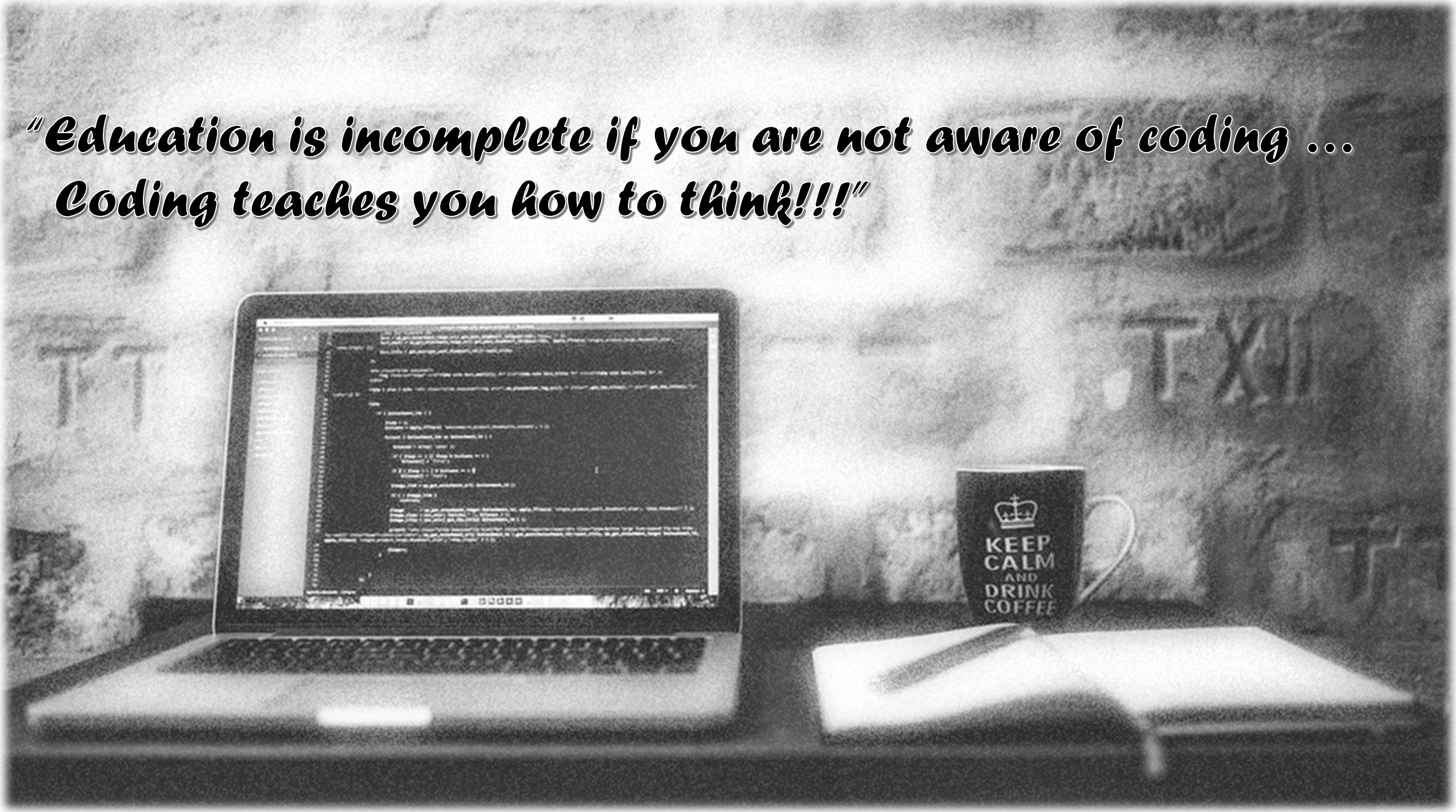
- // Math.random() returns a random number between 0.0-0.99.
- double rnd = Math.random();

- // rnd1 is an integer in the range 0-9 (including 9).
- int rnd1 = (int)(Math.random()*10);

- // rnd2 is in the range 1-10 (including 10). The parentheses are necessary!
- int rnd2 = (int)(Math.random()*10) + 1;

- // rnd3 is in the range 5-10 (including 10). The range is 10-5+1 = 6.
- int rnd3 = (int)(Math.random()*6) + 5;

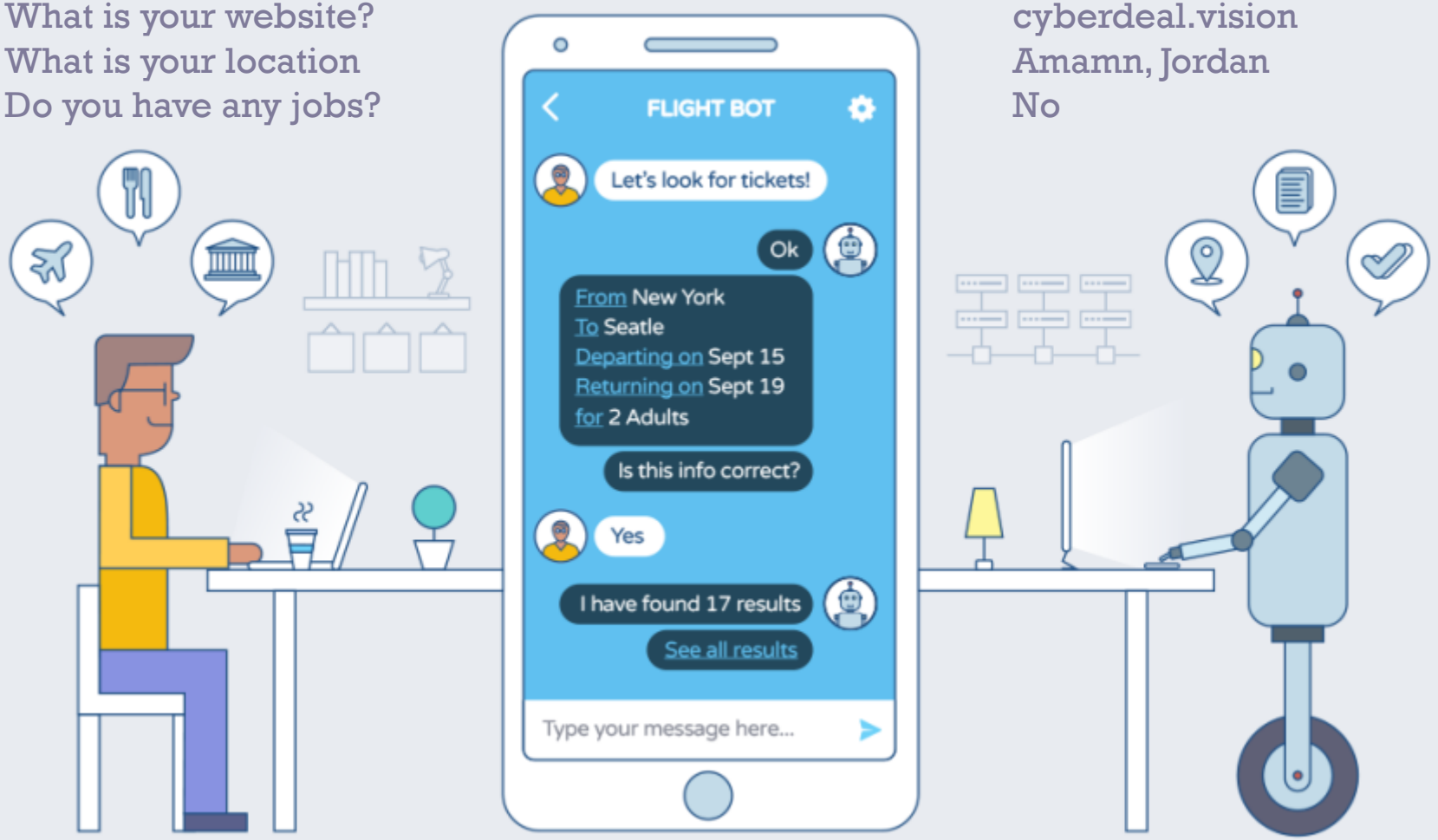
"Education is incomplete if you are not aware of coding …
Coding teaches you how to think!!!"



Build a simple chatbot

What kind of food do you sell here?
What is your website?
What is your location
Do you have any jobs?

Indian, Chinese, Arabic
cyberdeal.vision
Amamn, Jordan
No



Build a simple chatbot

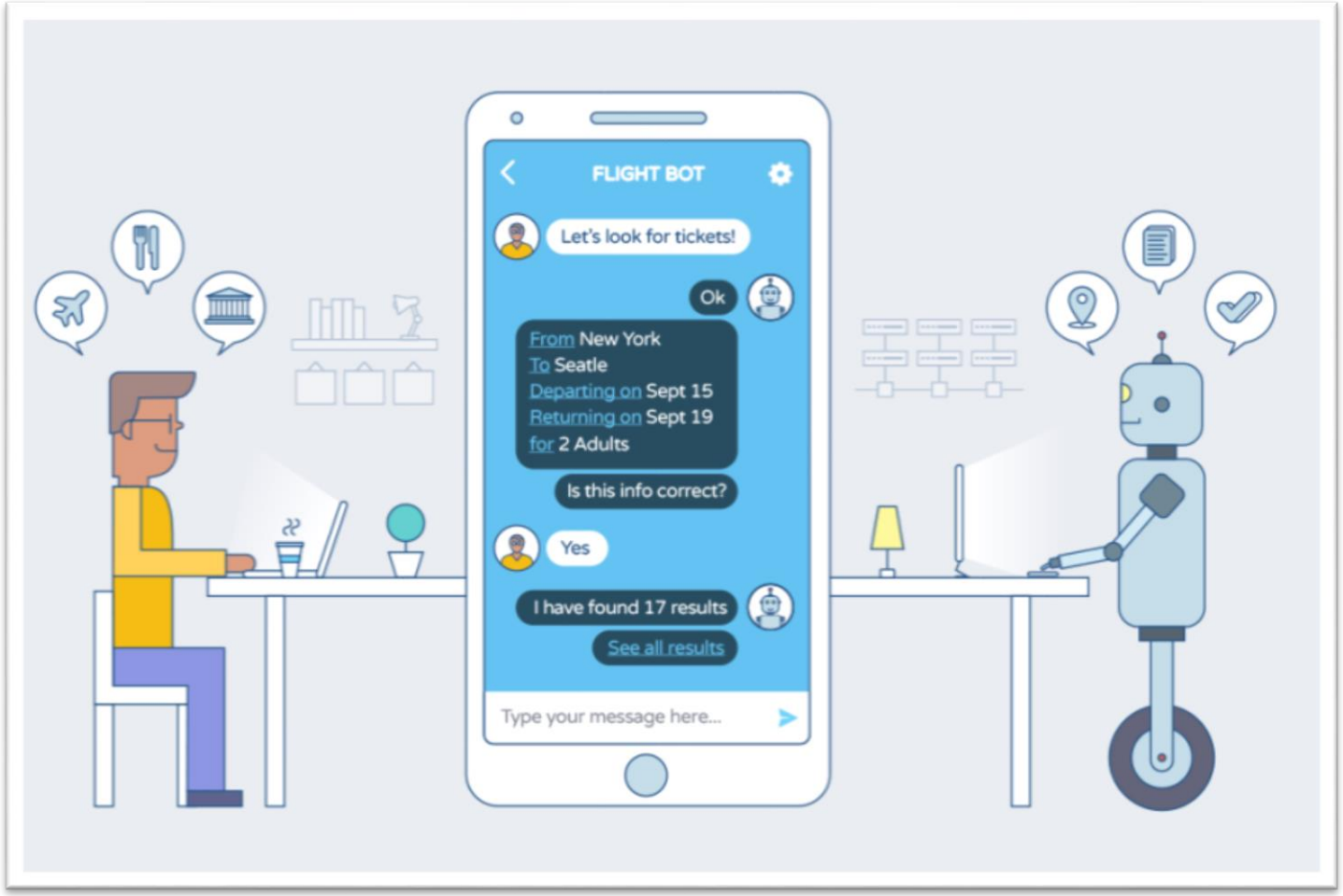
Hello Yaman, How can we help you today?

- What kind of food do you sell here?
- What is your website?
- What is your location
- Do you have any jobs?

* Indian food
- Chicken Tikka (4 JD's)
- Butter Chicken (4.5 JD's)
- Biryani (5 JD's)
* Arabic Food
- Mansaf
- Kabseh
- Mlokhya

Cyberdeal.vision

Amman, Jordan



Challenge:
Program a simple chat-bot for any sector you like
(health, restaurants, sports, business, TV, travel)
Basically the user can ask for help, and the program will answer his questions

{ MAX & SECOND MAX }

```
      printf( "Enter next number: " ); /* prompt for next number */
      scanf( "%d", &number );

      /* if current number is greater than largest */
      if ( number > largest ) {

         /* update second largest with previous largest */
         secondLargest = largest;

         /* update largest with current number */
         largest = number;
      } /* end if */
      else {

         /* if number is between secondLargest and largest */
         if ( number > secondLargest ) {
            secondLargest = number;
         } /* end if */

      } /* end else */
```

```
 4  int number = 373;
 5  int lastDigit , tempNumber = number;
 6  int reverse=0 , armstrong=0 , counter=0;
 7
 8  while (tempNumber!=0)
 9      {
10      lastDigit = tempNumber % 10;
11
12      reverse = reverse * 10 + lastDigit;
13
14      armstrong += (lastDigit*lastDigit*lastDigit);
15
16      tempNumber = tempNumber / 10;
17
18      counter++;
19      }
```

This code solves the palindrome & Armstrong number problem (note that the Armstrong only works for 3 digits), also counts the number of digits …

In our example, the number is ‘373’

Note that I stored the original number in a temp variable so I could compare the original number to other numbers later…

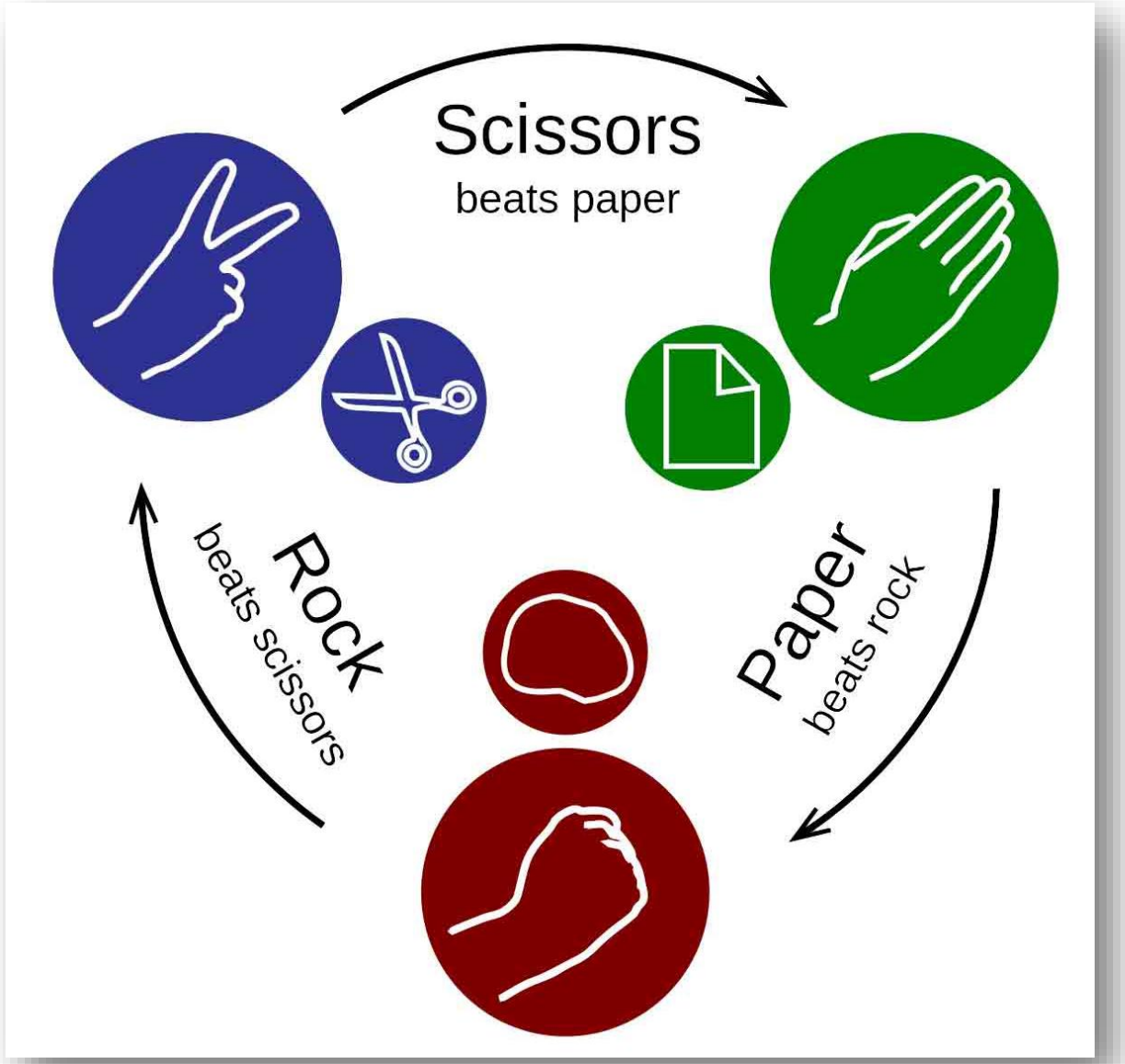
‘Line 10’ we separated the last digit from the right (373 % 10 = 3)

‘Line 12’ is used to store the original number reversed (e.g. 123 becomes 321)

‘Line 14’ is used to store the cube of each digit and sum it up …

(e.g. 153 is a Armstrong because
((1*1*1) + (5*5*5) + (3*3*3) = 153))

‘Line 16’ chops off the last digit
(373 / 10 = 37)

‘Line 17’ counts how many time the number chopped off a digit…

Rock Paper Scissors Game