UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

ECE 150 *Fundamentals of Programming*

Repetitious operations

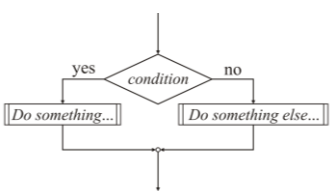
Prof. Hiren Patel, Ph.D.
Douglas Wilhelm Harder, M.Math. LEL
hdpatel@uwaterloo.ca dwharder@uwaterloo.ca

Outline

- In this presentation, we will:
  - Review conditional statements
  - Introduce repetitious operations
    - Review the game of Pictionary™
  - Describe how these solve specific real-wold problems
  - Understand the need for true/false conditions to continue looping
  - Look at the flow chart

Conditional statements

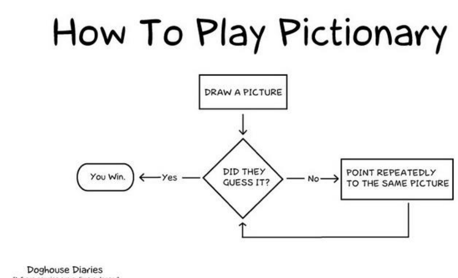
- In real-world situations, you take actions based on criteria or conditions
  - Programming languages restrict decisions to only those that are absolutely necessary to accomplish goals
  - This boils down to Boolean-valued conditional statements




```
if ( condition ) {
    // Do something…
} else {
    // Do something else…
}
```

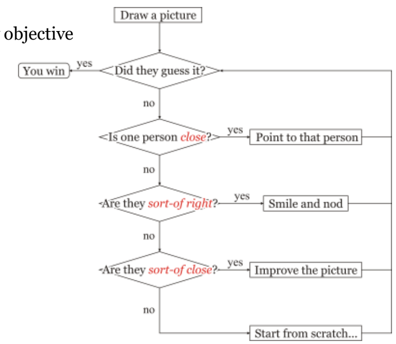

Repetitious operations

- Another possibility is to repeat an operation until some goal is accomplished
  - In Pictionary, you are given a word and you must get the audience to guess the word based on a drawing



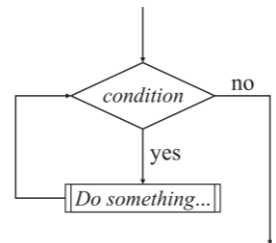
Repetitious operations

- A more realistic version followed by many players
  - Each decision is not really objective
  - What is "close"?
  - What is "sort-of"?



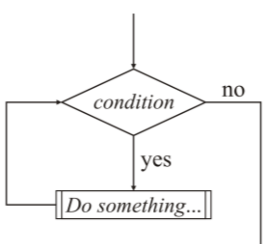
Flow charts

- A *flow chart* for simple repetition is to continue repeating a code block as long as some condition is TRUE
  - As soon as the condition is FALSE, we proceed to other statements



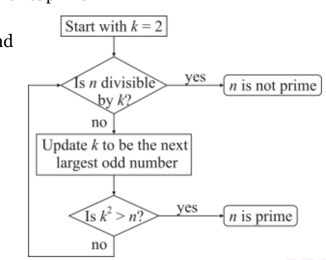
Flow charts

- The repetitive behavior of this repetitive flow chart component leads to the name *looping statement*
  - We continue *looping* until a condition fails



Repetitious operations

- In mathematics, you learned a number of algorithms that depended on repetition:
  - To determine if n is prime:
    - Try dividing n by every integer $k = 2, \ldots, n - 1$
    - If any remainder is zero, then n is not prime
  - Better yet, try dividing n by 2 and every odd number up to $\sqrt{n}$



Repetitious operations

- Even long division is a repetitive algorithm:
  - Let m and n be real numbers both of which have a finite number of digits
  - Multiply both m and n by the smallest power of 10 that allows n to be an integer
  - In calculating $m \div n$, perform the following:
    - If $|m| < 1$, let the solution be the decimal point "." and let r be the first digit after the decimal point
    - Otherwise, let r be the first non-zero digit of m and let the solution be ""
  - Now iterate:
    - Find the largest integer multiple d of n such that $r - dn \geq 0$, append d to the solution and let $r - dn$ be the new r
    - Add the next digit of m to the end of the new r and if the next digit is the first digit after the decimal place, place a decimal point at the end of the solution

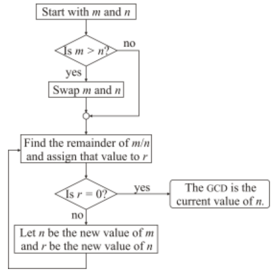
Repetitious operations

- In mathematics, you learned a number of algorithms that depended on repetition:
  - To calculate $n!$
    - You multiplied together all of the integers $k = 2, \ldots, n$
  - To calculate $\binom{n}{k}$, you find $\frac{n!}{(n-k)!k!}$ which requires three factorial calculations followed by one long division
    - You can reduce your work by calculating $n(n-1)(n-2)\cdots(n-k+1)$ and then dividing by $k!$

Repetitious operations

- Each of these repetitious operations requires an exact number of steps to be performed
  - In some cases, you don't know *a priori* when to stop
  - Consider finding the greatest common divisor (GCD) of 3094 and 19635:

$$19635 - 6 \times 3094 = 1071$$
$$3095 - 2 \times 1071 = 952$$
$$1071 - 1 \times 952 = 119$$
$$952 - 8 \times 119 = 0$$



  - The GCD is therefore 119

Repetitious operations

- In programming languages, it is also necessary to repetitively perform a sequence of actions
  - In Pictionary, the choice of action was someone fuzzy
  - In mathematics, in each case the repetitive operation is well defined
  - In some cases, we know how often we will have to repeat an operation, in others, we may not know until we get a solution
    - That is, if a solution even exists
- Like decision making, in programming languages we will have to reduce the decision as to how often to repeat down to a simple yes-no question

Summary

- In this presentation, you now
  - Are familiar with repetitious operations
  - Have re-familiarized yourself with such operations in secondary school
  - Know the flow chart for such operations
  - Understand that the conditions for looping must be Boolean-valued
    - Either true or false

References

[1] Wikipedia
https://en.wikipedia.org/wiki/Pictionary

Acknowledgments

Proof read by Charlie Liu.

Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.



Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.