Sept 3, 2024

* Welcome
* Introduction: class website: cs.brynmawr.edu → cmsc245
* Warm-up Exercise
* The World of Programming languages
  - Read articles posted by Dijkstra, Hayes, Kumar.

Warm up Exercise

What we do:

Problem → Solution (Algorithm) → Pseudocode → Program

Example

Problem: Greatest Common Divisor (GCD)

Definition: $gcd(a, b) = c$, $a, b \neq 0$

c is the largest number that divides both a and b

e.g. $gcd(9, 15) = 3$

1, (3), 9

1, (3), 5, 15

$gcd(54, 24) = 6$

1, 2, 3, (6), 9, 18, 27, 54

1, 2, 3, (6), 8, 12, 24

(c 300BC) **Euclid's Algorithm:** To compute the gcd(a, b) check to see if a and b are equal. If they are, then either a or b is the answer. O/w replace the larger of a and b with their difference. Repeat.

e.g. $gcd(9, 15) = gcd(9, 15-9) = gcd(9, 6)$
$= gcd(9-6, 6) = gcd(3, 6)$
$= gcd(3, 6-3) = gcd(3, 3) = 3$

Pseudocode

```
function gcd (a, b)
    while a ≠ b do
        if a > b then
            a ← a − b
        otherwise
            b ← b − a
    return a.
```

Program Code – Procedural

```
int gcd (int a, int b) { //compute gcd of a & b
    while (a != b) {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
} //gcd()
```

In Java, C, C++

```
def gcd(a, b):
    while a != b:
        if a > b:
            a = a - b
        else:
            b = b - a
    return a
```

Python

indentation

Note: differences in syntax

Alternately – Pseudocode

functional

```
function gcd(a,b)
    if a = b then
        return a
    otherwise if a > b then
        return gcd(a-b,b)
    otherwise
        return gcd(a,b-a)
```

e.g. gcd(9,15) = gcd(9,6)
= gcd(3,6)
= gcd(3,3)
= 3

Java/C/C++

```
int gcd(int a, int b) {
    if (a==b)
        return a;
    else if (a > b)
        return gcd(a-b,b);
    else return (gcd(a,b-a);
}//gcd()
```

Q. What programming languages can we name?

Java 1995	Fortran 1957	COBOL 1959	SmallTalk 1972	Prolog 1972	
C 1972	LISP 1958	BASIC 1964	Eiffel 1985	ML 1973	Rust 2015
C++ 1980-83	ALGOL 1958	PASCAL 1970	Icon ~1977	Forth 1970	Sisal 1983
C# 2001	PL/1 1964	ADA 1983	AWK 1983	CommonLisp 1984	Ruby 1995
Python 1991	Perl 1987	PHP	Haskell 1990	GO 2009	JS 1995
Scala 2003	Kotlin 2011	Swift 2014	R 1993		Lua 1993

Q. How many PLs are there?

300 - 3000! depending on who you ask.

Q: What is a programming language?

Not HTML, CSS, REACT, etc.
Why?

Q. What are the top ten PLs today?

See the TIOBE index:

Python	Matlab
C++	Delphi/Pascal
C	PHP
Java	RUST
C#	RUBY
JS	Swift
SQL	Assembly
VBasic	Kotlin
GO	R
Fortran	Scratch

Q. Why are there so many programming languages?

For next class:

Read Dijkstra, Hayes, Kumar articles