CSE-115: Structured Programming Language

Lecture 1

Introductory Lecture

CSE-115

- Course Code: **CSE-115**
- Course Title: Structured Programming Language
- Course Teacher: **Dr. Ashikur Rahman**

  Associate Professor, Dept. of CSE, BUET
- Credit: **3**

My Brief Profile

- Dr. Ashikur Rahman

  Associate Professor, CSE, BUET.

  B.Sc.: BUET, 1998

  M.Sc.: BUET, 2001

  Ph.D.: University of Alberta, Canada, 2006

  Postdoc: University of Calgary, Canada, 2011

  State University of New York, USA, 2012

Syllabus

- constant, variable and data types,
- operator and expression, type conversion,
- decision making, branching and looping,
- arrays and strings,
- user defined functions,
- structures and union, bit field and bit-wise operations,
- pointer,
- file management in C,
- dynamic memory allocation and linked list

Reference Book

- Text:
  - Teach Yourself C (3rd Edition) – Herbert Schildt
  - Programming in ANSI C – Balagurusamy
- Other Reference Book:
  - C – How to Program (4th Edition) – Deitel & Deitel

Learning Style

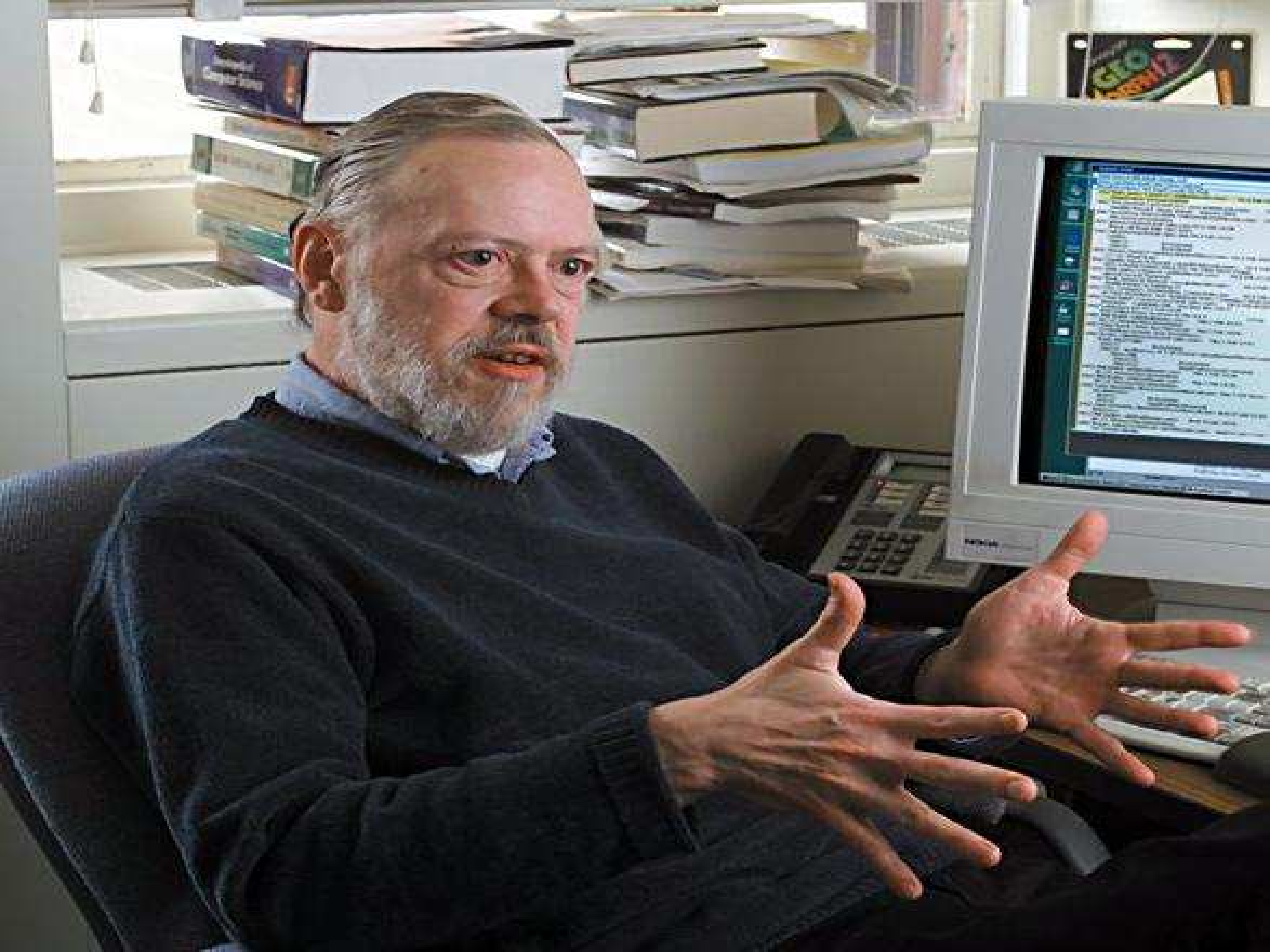
- This course is practical oriented.
- Three key techniques to perform better in this course:

(1) Practice
(2) Practice
and (3) Practice

  - (because practice makes a man perfect ☺)
- Memorization will not help you to get a good result.
- Copying Code:
  - Strictly prohibited.
  - Will be severely punished if you are caught.

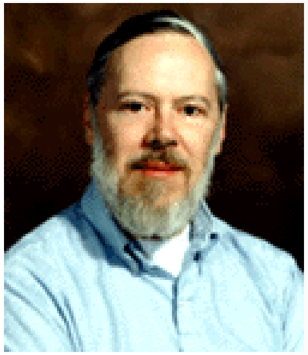
Who is the inventor of C Programming Language?

Getty Images



C Brief History

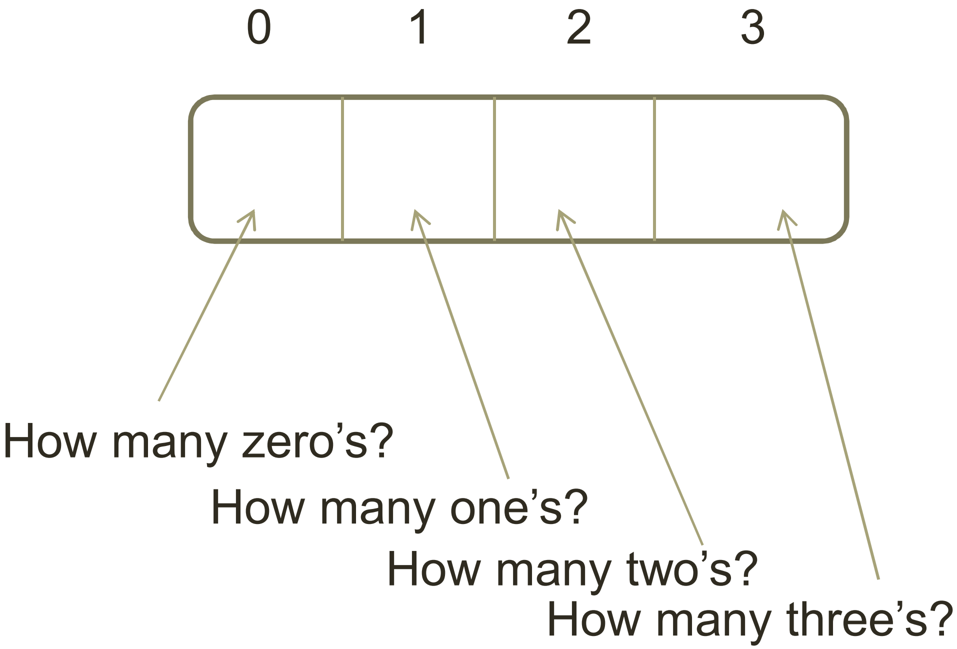
- Developed by Dennis Ritchie at AT&T, early 70s, for DEC PDP-11
- Unix written in, closely associated with, C
- Family of languages:
  - BCPL, Martin Richards
  - B (typeless), Ken Thompson, 1970
  - C, Dennis Ritchie, Bell Labs, early 70s
  - C++, Bjarne Stroustrup, Bell Labs, 80s
  - Java, James Gosling Sun, 1995
  - C#, Microsoft, recently
- C++, Java, C# conserve (much) C syntax



Why should we learn programming?

Helps us to solve many, many, many, many, …. interesting, useful and/or complex problems

4-digit number problem:



How many zero’s?

How many one’s?

How many two’s?

How many three’s?

Birthday Problem:

1 3 5 7
9 11 13 15
17 19 21 23
25 27 29 31

2 3 6 7
10 11 14 15
18 19 22 23
26 27 30 31

4 5 6 7
12 13 14 15
20 21 22 23
28 29 30 31

8 9 10 11
12 13 14 15
24 25 26 27
28 29 30 31

16 17 18 19
20 21 22 23
24 25 26 27
28 29 30 31

3 and 8 are good enough!

14 = 3+3+8

15 = 3+3+3+3+3

16 = 8+8

How to solve all these interesting problems?

But computer can only understand 0’s and 1’s!

- Computer’s language
  - 0’s and 1’s
  - Machine language
  - Hard to code for human beings
- What’s the solution then?

  - Develop English like-languages
    - High-level languages like C
  - Let compiler translate at the background

A Simple C Code

```
#include <stdio.h>

int main()
{
 printf(“Welcome to CSE 115”);
  return 0;
}
```

How to Run

- Save a source code with extension “c” or “cpp”. (Ex: **first.c**)
- Compile it and link it
  - Output: first.exe
- Run the program.
- Output of the program:

  -
    ```
    Welcome to CSE 115
    ```