



# CS 103 Computer Programming Spring 2018

By Faiza Sattar

[faiza.sattar@nu.edu.pk](mailto:faiza.sattar@nu.edu.pk)

## Week 01

### Introduction to the Course

22<sup>nd</sup> /Jan – 26<sup>th</sup> /Jan

# Today's Outline

- Administrative Stuff
- Grading Policy
- Overview of CS-103
- Background Knowledge (Pre-Requisite)
- Course Outline
- Books and reference material
- Generations of Languages
- Introduction to Object Oriented Paradigm
- Review

3

## Administrative Stuff

### About Me:

- Graduated in 2010 from NEDUET
- Completed Masters in 2015 from NEDUET
- Associated with FAST since 2013
- Courses taught:
  - EE 213 Computer Organization and Assembly Language
  - CS 103 Computer Programming
  - CS 309 Object Oriented Analysis and Design
- Labs Conducted:
  - EL 213 Computer Organization and Assembly Language
  - EL 227 Digital Logic Design
- Research interests:
  - Software engineering
  - Database systems/Data Mining
  - Image Processing

## Administrative Stuff (Contd.)

## Consultation hours

- Office : Room 14
- Office hours:  
Monday, Tuesday (2-3)

## Tentative Grading Policy

- |  |      |
|--|------|
| • At least 5 Assignments                                 | 10%  |
| • At least 5 Quizes <small>(surprised/announced)</small> | 10%  |
| • Mid-Term   | 30%  |
| • Final  | 50 % |

Start Scoring....

Best of Luck!

## Attendance

Take it serious!

## About the Course

- Objective of this course is:
  - to make students familiar with the concepts of **object-oriented programming**
  - to model real world situations
  - to program in OOP paradigm
- Concepts will be reinforced by their implementation in **C++**
- *A strong class participation from the students will be encouraged.*

## Background Knowledge

- You must be able to write a program in some language.
- Understand variables, control structures, functions/subroutines.
- Procedural Programming

## Course Outline

- Procedural vs Object Oriented Programming
- Object-Orientation
- Objects and Classes
- Overloading
- Inheritance
- Polymorphism
- Generic Programming
- Exception Handling

# Books

## Text Books

- 1- "Problem Solving with C++", 9e Global Edition,  
By Walter Savitch, ISBN13:9781292018249, Addison-Wesley, 2015
- 2- C++ How to program  
By Deitel & Deitel

## Reference Books

- 1- The C++ Programming Language  
By Bjarne Stroustrup.
- 2- Object Oriented Software Engineering  
By Jacobson, Christerson, Jonsson, Overgaard

What is a Programming  
Language?

## What is a Programming Languages

- A programming language is **a set of rules** that provides a way of telling a computer what operations to perform.

## What is a Programming Language

- A programming language also has words, symbols and rules of grammar.
- The grammatical rules are called **syntax**.
- Each programming language has a different set of syntax rules.

## What Are the Types of Programming Languages

- First Generation Languages
- Second Generation Languages
- Third Generation Languages
- Fourth Generation Languages
- Fifth Generation Languages

## First Generation Languages

- Machine language
  - **Operation code** – such as addition or subtraction.
  - **Operands** – that identify the data to be processed.
  - Machine language is **machine dependent** as it is the only language the computer can understand.
  - Very **efficient** code but very difficult to write.
  - Code cannot be ported to other systems and has to be rewritten

Code Example:

```
10101010011000101  
10011010100000010  
1111111101000101
```



## Second Generation Languages

- Assembly languages
  - **Symbolic** operation codes replaced binary operation codes.
  - Assembly language programs needed to be “**assembled**” for execution by the computer. Each assembly language instruction is translated into one machine language instruction.
  - Very **efficient** code and easier to write.
  - Code cannot be ported to other systems and has to be rewritten

### Code Example

```
LDA A
ADD #5
STO A
JMP #3
```

## Third Generation Languages

- 3GLs are the first to use English like phrasing, making them **easier** to use than the previous languages.
- 3GLs are **portable**, means the object code created for one type of system can be translated for use on a different type of system.
- The translation of source code to object code is accomplished by a machine language system programs called **compiler/interpreter**.

## Third Generation Languages

- The following languages are 3GLs
  - FORTRAN
  - COBOL
  - BASIC
  - Pascal
  - C
  - C++
  - Java

```
public boolean handleEvent (Event evt) {  
    switch (evt.id)  
    {  
        case Event.ACTION_EVENT:  
        {  
            if ("Try me" .equals(evt.arg)) {...
```

## Fourth Generation Languages

- A high level language (4GL) that requires fewer instructions to accomplish a task than a third generation language.
- Used with databases
  - Query languages
  - Report generators
  - Forms designers
  - Application generators

```
EXTRACT ALL CUSTOMERS WHERE "PREVIOUS PURCHASES"  
TOTAL MORE THAN $1000
```

## Fifth Generation Languages

- Programming language based on constraints given to the program rather than algorithms written by a programmer.
- Used mainly in artificial intelligence research.
- uses a visual or graphical development interface to create source language
- Prolog, mercury are the best known languages as 5GLs.

## Generations of Languages

Generation	Classification
1st	Machine languages
2nd	Assembly languages
3rd	Procedural languages
4th	Application languages (4GLs)
5th	AI techniques, inference languages

# Levels of Programming Languages

High-level program

```
class Triangle {  
    ...  
    float surface()  
        return b*h/2;  
}
```

Low-level program

```
LOAD r1,b  
LOAD r2,h  
MUL r1,r2  
DIV r1,#2  
RET
```

Executable Machine code

```
0001001001000101001001  
001110110010101101001.  
..
```

## References

- Virtual University of Pakistan
- Books – Dietal & Dietal
- OOP By Robert Lafore
- Various other web references

