# Introduction to Object-Oriented Programming

# Contents

- Brief history of computer programming
- Procedural programming
- Object-oriented programming

# Computer Programming

- A computer program is a list of *instructions* that tell computer what to do
- Example of a simple PASCAL program:

```
Program Lesson1_Program1;
Begin
        Write('Hello World. Prepare to learn PASCAL!!');
        Readln;
End.
```

# Programming Languages

Three main categories:

- Machine languages
- Low-level assembly languages
- High-level programming languages

# Machine Languages

- Composed of 0 and 1
- Is the “native” language of a computer, but difficult to program
- Example of machine codes:

| Machine Instruction | Machine Operation |
|---|---|
| 00000000 | Stop Program |
| 00000001 | Turn bulb fully on |
| 00000010 | Turn bulb fully off |
| 00000100 | Dim bulb by 10% |
| 00001000 | Brighten bulb by 10% |
| 00010000 | If bulb is fully on, skip over next instruction |
| 00100000 | If bulb is fully off, skip over next instruction |
| 01000000 | Go to start of program (address 0) |

# Assembly Languages

- Computer instructions are represented in symbolic codes
- Needs to be translated into machine codes before processing
- Example of assembly codes:

```
mov     dx,msg2          ; print msg2
mov     cx,msg2len       ;
call    PrintString      ;
```

- Assembly language is a step towards easier programming

# High-level Languages

- Syntax is similar to human languages
- Need to be compiled into machine codes for executing
- Example of high-level codes:

```c
#include <stdio.h>
int main()
{
   // printf() displays the string inside quotation
   printf("C Programming");
   return 0;
}
```

- High-level language is a big step towards easier programming

# Classifying high-level languages

- Historically, high-level languages are divided into two categories:
  - Procedural Programming
  - Object-Oriented Programming (OOP)

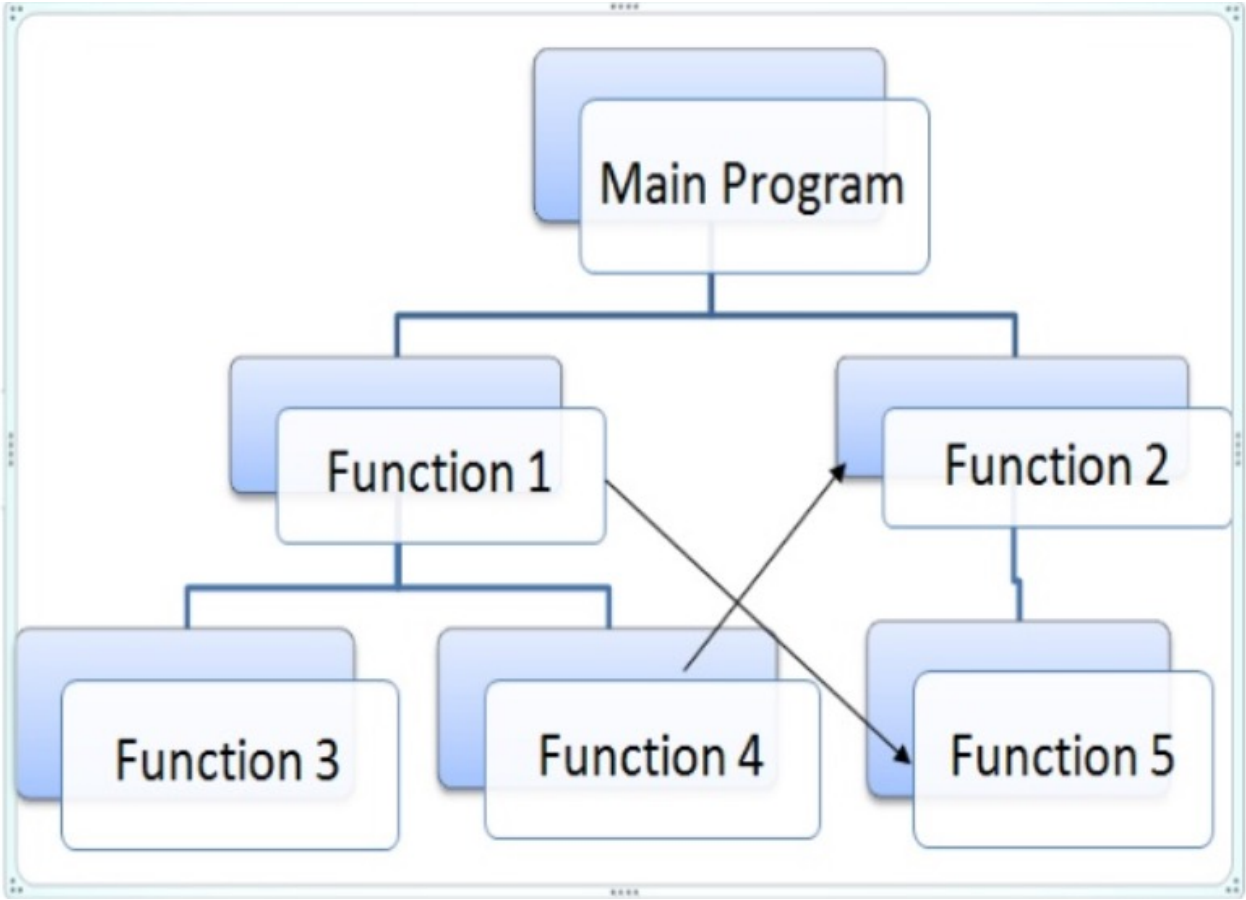
# Procedural Programming

- Procedural programming is a programming paradigm where program contains a sequential sets of computational/linear commands to be carried out by the computer

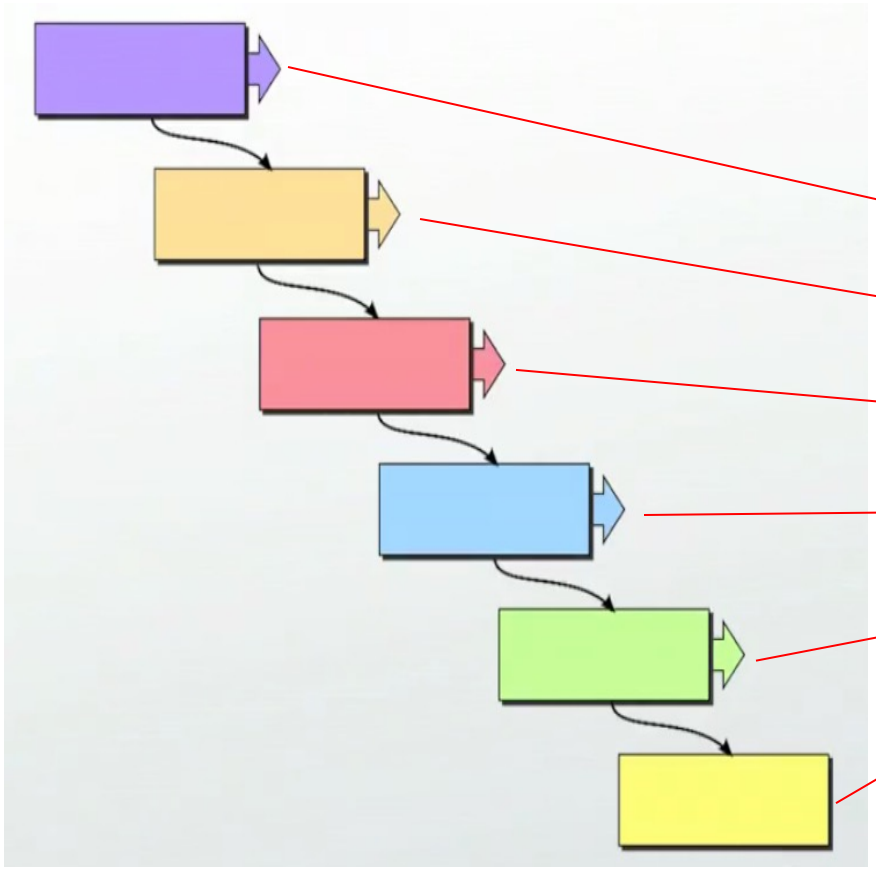
```c
#include <stdio.h>
int main()
{
    int a, b, sum;

    printf("Enter two integers: ");

    scanf("%d %d",&a, &b);

    sum = a + b;

    printf("%d + %d = %d", a, b, sum);

    return 0;
}
```

# Procedural Programming

- In procedural programming, computer program is divided into small parts called functions



# Example of Procedural Programming



```c
#include <stdio.h>
int main()
{
    int a, b, sum;

    printf("Enter two integers: ");

    scanf("%d %d",&a, &b);

    sum = a + b;

    printf("%d + %d = %d", a, b, sum);

    return 0;
}
```

Six Sequential Computation Steps

# Procedural Languages

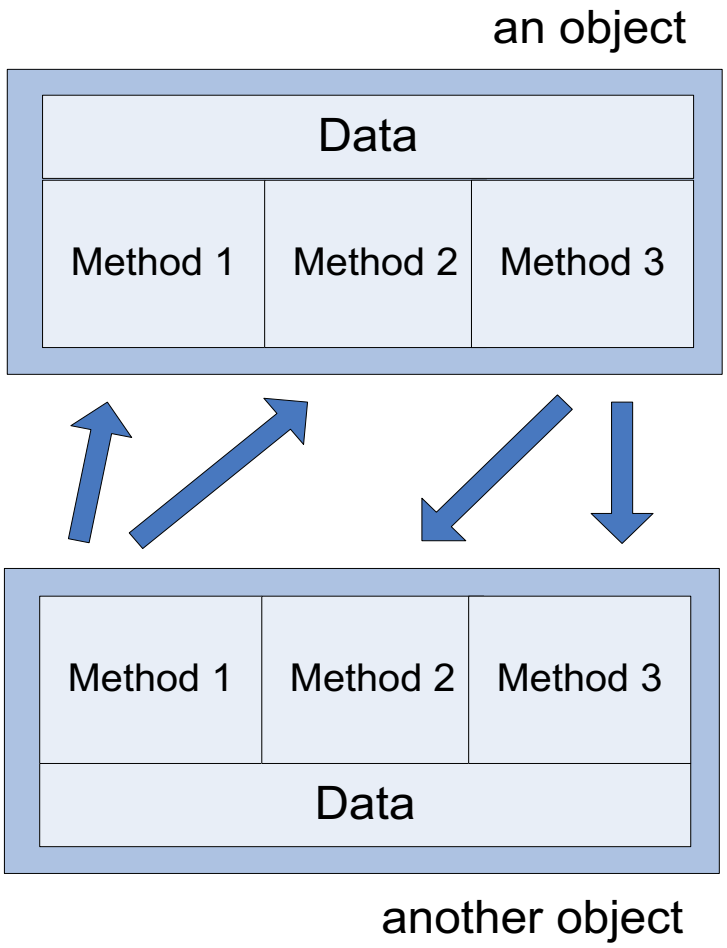
COBOL PROGRAMMING

C Language

# Object-Oriented Programming

- OOP is a programming paradigm where computer program is divided into parts called objects
- Key idea:

  *"The real world can be described as a collection of objects that interact"*

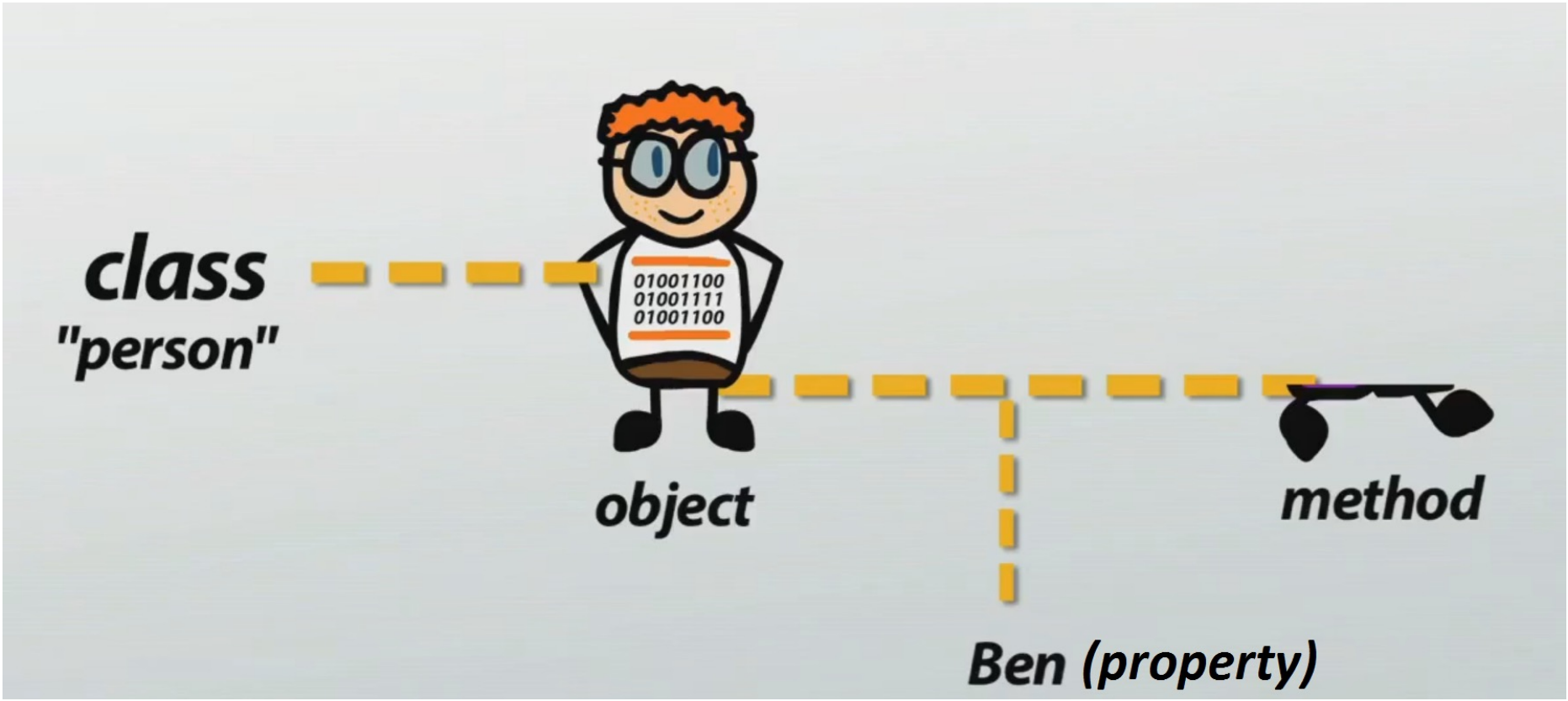


# Object-Oriented Programming

- In OOP, object is a “thing” that includes both *data (properties)* and *functions (methods)*

# OOP Languages

- In OOP languages, programmers create programs using “blueprints” of data models called classes



# Example of OOP Languages




C#

# Example of OOP Languages

- Java will be used as the language to demonstrate OOP concepts in this course