

# Course Information and Introduction

Arash Rafiey

August 22, 2017

## ① Instructors : Arash Rafiey

- Email : arash.rafiy@indstate.edu
- Office : Root Hall A-127
- Office Hours : Tuesdays 11:30 pm – 12:30 pm. Root Hall, A127.

## ② Course Home Page :

<http://cs.indstate.edu/~arash/CS256.html>

## Textbooks :

The C Programming Language, B. Kernighan, and D.M. Ritchie, Printice Hall PTR, Second Edition 2012.

- 1 Homework assignments (6 assignments each 4 % )
- 2 Midterm (26 % )
- 3 Final (50 % )

## Objective :

Learning C (C++) programming Language (up to certain level).

## Objective :

Learning C (C++) programming Language (up to certain level).

A programming language is a translator between human and machine.

Machine language is a set of codes each consists of a binary string usually of length multiple of 8 (00000101,1000000,...)

Since it is difficult to use binary codes we design programming languages.

Each program language has a compiler which creates a machine codes.

## Objective :

Learning C (C++) programming Language (up to certain level).

A programming language is a translator between human and machine.

Machine language is a set of codes each consists of a binary string usually of length multiple of 8 (00000101,1000000,...)

Since it is difficult to use binary codes we design programming languages.

Each program language has a compiler which creates a machine codes.

Learn how to program in C some of the simple algorithms in computer science.

## Objective :

Learning C (C++) programming Language (up to certain level).

A programming language is a translator between human and machine.

Machine language is a set of codes each consists of a binary string usually of length multiple of 8 (00000101,1000000,...)

Since it is difficult to use binary codes we design programming languages.

Each program language has a compiler which creates a machine codes.

Learn how to program in C some of the simple algorithms in computer science.

How does it look like to become a programmer ?



# Topics to cover

- 1 How to write a simple program (getting started)
- 2 Types, Operations, and Expression
- 3 Control Flow (**If Else** statement, **while** and **for** statements)
- 4 Functions and Program Structure (simple recursive functions )
- 5 Structures (complex data type, name and address and zip code is one complex data type )
- 6 Pointers and Array (hard to learn pointers!! )
- 7 Input and Output (Read from and write into a file on a disk)

# A Simple Program

```
# include <stdio.h> // allow us to use the standard library
main() // each program must have a main function
{
    printf("hello, world \n "); // \ is for a new line
}
```

We save the file (the hello world program) with extension `.c` file.  
For example `hello.c`

We save the file (the hello world program) with extension `.c` file.  
For example `hello.c`

In the UNIX operating system type

```
cc hello.c
```

The `c` compiler creates an object file (executable file) by default  
`a.out` .

We save the file (the hello world program) with extension `.c` file.  
For example `hello.c`

In the UNIX operating system type

```
cc hello.c
```

The `c` compiler creates an object file (executable file) by default  
`a.out` .

In order to run the program we type

```
./a.out
```

We save the file (the hello world program) with extension `.c` file.  
For example `hello.c`

In the UNIX operating system type

```
cc hello.c
```

The `c` compiler creates an object file (executable file) by default  
`a.out` .

In order to run the program we type

```
./a.out
```

It will print

```
hello world
```

If you want to give a different name then type

```
cc main.c -o main.out
```

Then your executable file would be main.out

Type ./main.out

# Reading from the command line

```
# include <stdio.h> // allow us to use the standard library
main() // each program must have a main function
{
    printf("hello, enter an integer number \n");
    int x;
    scanf("%d",&x);
    printf("%d", x);
}
```