

# LEARN PROGRAMMING

STUDY GROUP- SESSION #1

**Weekly: Wednesday 19:15 to 22:15**  
**R307 G29**

# WHO ARE WE?

- International FIN students.
- Masters in DKE/DE
- Have few years of experience with software development.
- Programming Language expertise C/C++, C#, Java, Python, R

## Volunteers

**Asema Hassan – MS(DE)**

**Rahul Jethwani – MS(DE)**

**Shadi Akhras – MS(DKE)**

**Axel Garcia – MS(DE)**

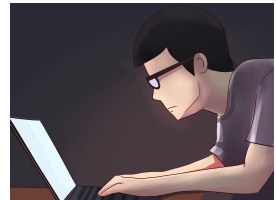
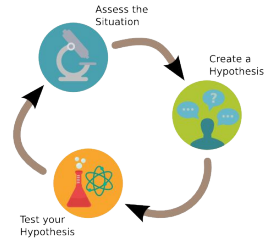
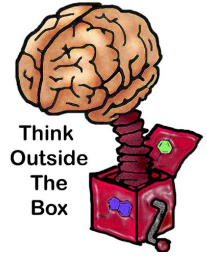
**Jatin Garg – MS(DKE)**

**Jawad Ahmad – MS(DE)**

**Kantha Raju – MS(DKE)**

# AGENDA

- **Think** out of the **box**.
- **Problem** solving **skills**.
- **Improve** your **analytical** thinking.
- **Learn** basics of **programming**.
- **Implement** your **solution**.
- **Practice** as much as you can!



# WHY DO WE NEED THIS GROUP?

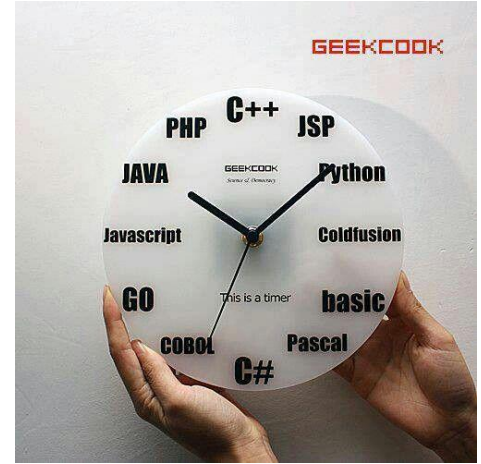
- Different background of studies.
- We belong to **Informatics** department.
- **Software Development** is the demand of today & tomorrow.
- No skills, **no benefit** of degree.
- **Means**, less than **no job** opportunities.

- Electrical
- Bio-Medical
- Mechanical
- Telecommunication
- Mechatronics
- Chemical etc..



# WHAT IS OUR PLAN?

- **Programming concepts**
- **Problem solving**
- **Pseudocode**
- **Flowcharts**
- **Algorithms**
- **Syntax**
- **Compiling**
- **Debugging**
- **Bug fixing**
- **PRACTICE..PRACTICE..PRACTICE**



# EXPECTATION?

## TODO:

- Discussions on concepts
- General problem solving
- Practicing new solutions
- Learning various languages
- Workshops/Weekly sessions

## NO:

- **NO SOLUTIONS** to COURSE ASSIGNMENT.
- **NO SOLUTIONS** to your project work.



# PROGRAMMING?

## 1. Why do we need to program at all?

A program is a **set of rules**, to give **instructions** to computer to perform certain **action**.

A **programming language** is a notational system for describing computation in a **machine-readable** and **human-readable** form.

# PROGRAMMING?

## 2. Language syntax?

English is a *natural language*. It consists of **words**, **symbols** and **grammatical rules**.

Each **programming language** also has words, symbols and rules of grammar called **syntax**.



# PROGRAMMING?

## 3. Types of Programming Languages?

**HIGH-LEVEL (C/C++, C#, Java, Python...)**

**LOW-LEVEL (Assembly)**

**EXECUTABLE MACHINE CODE (Binary)**

```
Class Person{  
    String name;  
    Int age;  
  
    String GetAddress(){  
        ....  
        Return address; }  
}
```

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

```
0010010010000101  
0010011101000001  
.....
```

# COMPUTER PROGRAM

Tells the computer:

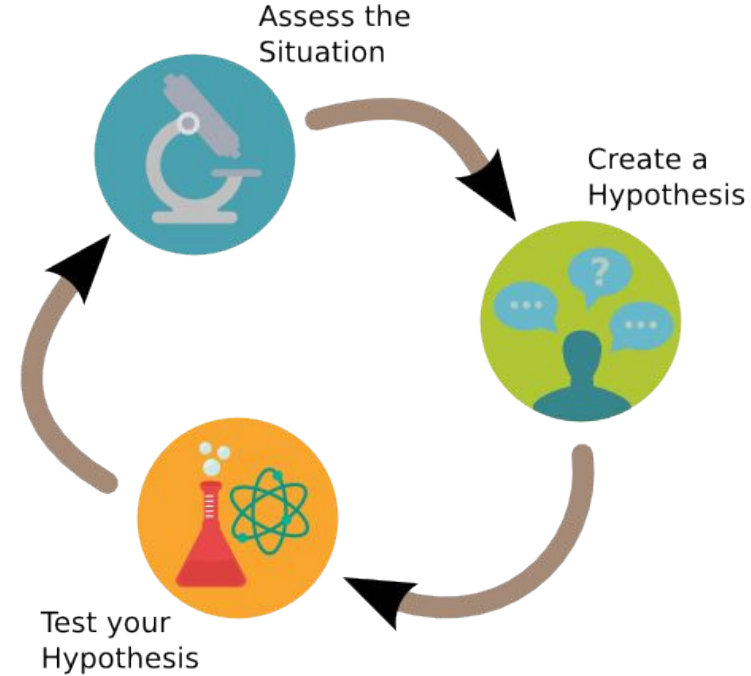
1. What **actions** you want the computer to perform
2. The **order** those actions should happen in

An effective program therefore needs:

1. A thorough understanding of the **problem**
2. A well **thought-out, step-by-step** solution to the problem

# PROBLEM SOLVING

1. Identify problem
2. Make a plan to solve problem
3. Divide problem into sub-tasks
4. Solve each tasks, step-by-step
5. Combine solution of sub-tasks
6. Test the solution
7. Does it solve the problem? YES/NO
8. End



# PROBLEM SOLVING

Developing an **Algorithm** is really just a type of Problem Solving.

-We have to:

**READ** and understand the problem

**THINK** about different solutions to the problem

**DESIGN** an approach that will solve the problem

**IMPLEMENT** that design

**TEST** to see if it works



# PSEUDOCODE

A simple human **readable** notational language to solve a problem. *\*You are describing a logic plan to develop a program, you are not programming\**

## RULES:

1. Write only one statement per line
2. Capitalize initial keyword
3. Indent to show hierarchy
4. End multi-line structures
5. Keep statements language independent

Keywords: **READ, WRITE, IF, ELSE, ENDIF, WHILE, ENDWHILE**

# PSEUDOCODE (EXAMPLE)

## Example 1:

**READ** name, grossPay, taxes

**IF** taxes > 0

**net** = grossPay – taxes

**ELSE**

**net** = grossPay

**ENDIF**

**WRITE** name, net

## Example 2:

**count** = 0

**WHILE** count < 10

**ADD** 1 to count

**WRITE** count

**ENDWHILE**

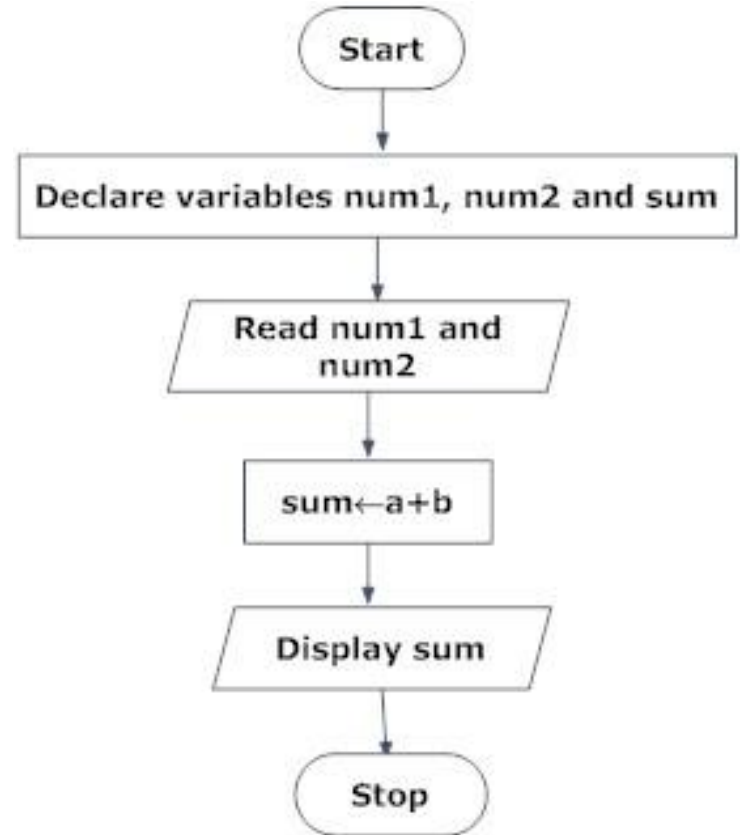
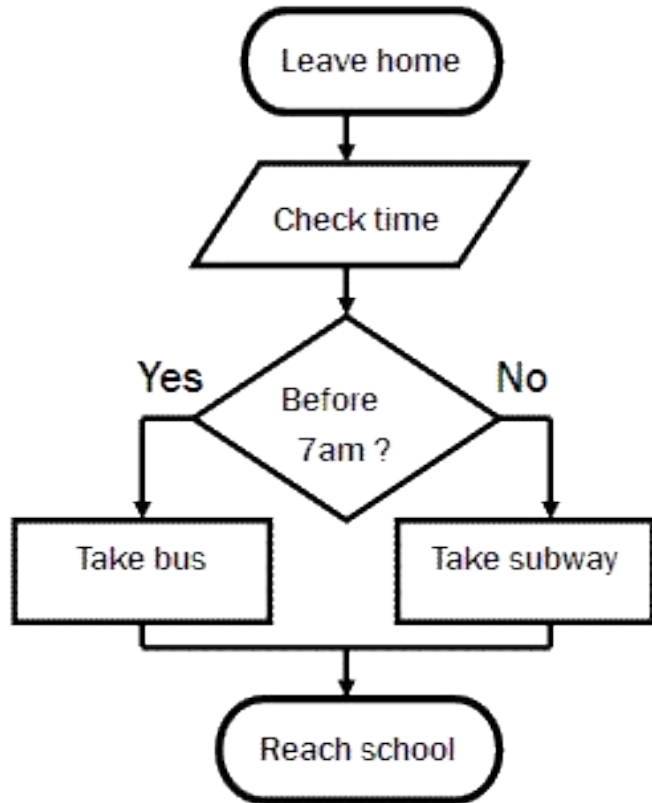
**WRITE** *“The End”*

# FLOW CHART

A type of diagram that **represents** an algorithm, workflow or process, showing the **steps** as boxes of various kinds, and their order by **connecting** them with arrows.

This diagrammatic representation illustrates a **solution model** to a given **problem**.

# FLOW CHART (EXAMPLE)





# TOOL & TECH

JETBRAINS: <https://www.jetbrains.com/student/>

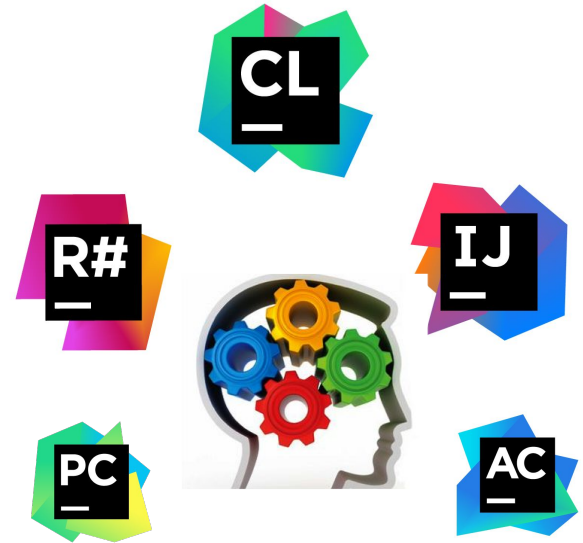
CLION - C/C++

Resharper - C#

IntelliJ IDEA - JAVA

PyCharm - Python

AppCode MacOSX/iOS



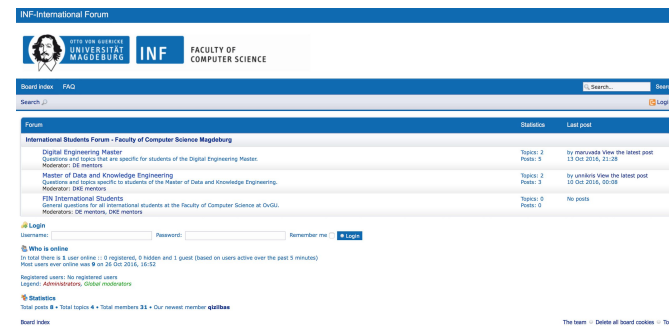
# ARE YOU COMMITTED?

- Dedicated...Motivated?
- Weekly meetups (7-10 PM)
- Programming practice
- Small talks about concepts
- New tasks in teams



# HOW TO REACH US?

- FORUM for International students?
- Link: <https://iif.cs.uni-magdeburg.de/index.php>
- “Study group programming”
- Login with your university ID.
- Post your questions on FORUM.
- Moderators will reply.



OFFICIAL COMMUNICATION ---> FORUM ONLY

# DISCIPLINE

- **Setup** room for study group.
- **Rearrange** before you leave.
- Keep it **clean**.
- Don't make **noise**.
- Keep **check** of who is coming in FIN.
- Don't **leave** doors open at night.

THANK YOU  
LET'S GET STARTED



# TASK FOR TODAY?

1. University grading system, taking courses data as input
2. A simple calculator for ADD,SUB,MUL,DIV, taking input from user.
3. A perfect SUNDAY plan;
  - a. Get ready
  - b. Leave house with family
  - c. Goto ATM, get cash
  - d. Goto a park/beach/movie theater

Show the process how a user will do it?

4. Make a cup of tea.