Adrian MILITARU

# Computer Organization project

## Project Expectations

## FIND TEAMMATES

You should form teams of 4-6 students

## ABOUT THE APP

How complex and what to include?

## SUGGESTIONS

A brief overview of what you can do

## DISCUSSIONS

# NEXT WEEKS...

Weeks 3 - 4

**Intro and requirements**

Weeks 5 - 6

**Suggestions, starting the project**

Weeks 7 - 8

**1st milestone and documentation guidelines**

Weeks 9 - 10

**Analyzing and visualizing scores**

Weeks 11 - 12

**Intermediary status**

Week 13

**Final presentation**

# Teammates

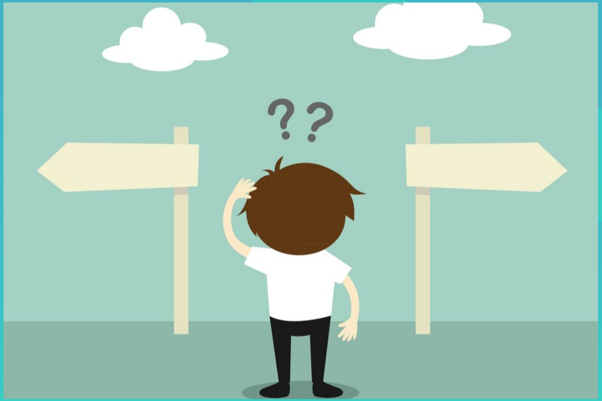
- Discuss about responsibilities inside team and also what you intend to do for the project

# Project

- Create a micro-benchmark app
- Test one or more functionalities of one or more **hardware component(s)** found in modern day PCs

# You decide!

- **Component**(s) being tested (e.g. CPU, hard disk, graphics card)
- **Functionality**(-ies) being tested (e.g. fixed point arithmetics, sequential read speed)
- **Algorithm**(s) used for benchmarking
- Performance measurement **criteria**



# Types of testing

## (some of them)

- **Testing is as important as any other part of the process!**
- Black-box vs. white-box testing
  - Advantages/disadvantages for each one of them?
- ***New!*** Grey-box testing
- Testing levels
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing

# Types of testing

## (some of them)

- Smoke testing
- Regression testing
- Alpha and Beta testing
- Continuous testing → **Automated tests**
- Destructive testing → **Exceptions, failures, worst case scenarios**
- Performance testing

# Performance testing

- Load testing → Scalability
- Stress testing → Extreme workloads
- Endurance testing → Expected load for a long period of time
- Spike testing → Sudden large spikes

# Suggestions of applications

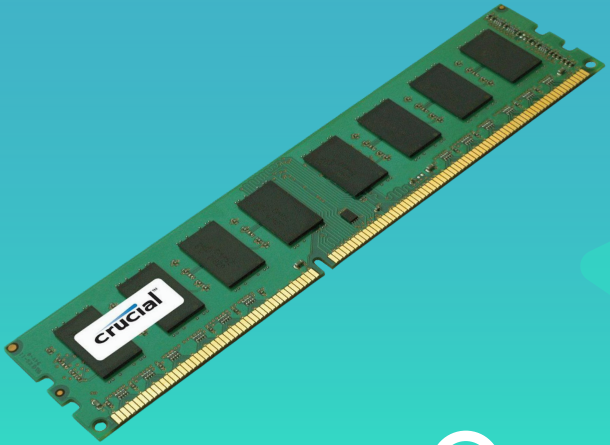
## CPU testing

- Fixed point arithmetics: conditional branching, array operations, recursivity, function calls, dhrystone etc.
- Floating point arithmetics: encryption, digits of Pi, convolution, image processing, fast fourier transform, trigonometry, whetstone etc.
- Threading, clock, caches & interprocess communication: recursivity, square root estimation, producer-consumer, hashing, cryptography, data access.
- Prime 95, Intel Burn Test, Orthos, Mandelmark, SuperPi, wPrime, others

# Suggestions of applications

## RAM testing

- Integrity of written versus read data
- I/O Throughput
- MaxxMEM, MemTest, others



# Suggestions of applications

## HDD testing

- Read versus write speeds
- File size speeds
- Sequential versus random access
- Block size allocation
- HD Tune, ATTO Disk Benchmark, CrystalDiskMark, Anvil Storage Utilities

# Suggestions of applications

## GPU testing

- Solutions for testing GPUs
- OpenGL/CL/DirectX: 3D rendering, FPS measurement (using Unity, Unreal Engine)
- Memory: textures, CUDA access time
- Interprocess: CUDA multithreaded algorithms, hashing, cryptography
- 3D Mark 11 (DX11), 3D Mark Vantage (DX10), 3D Mark 06 (DX9), Unigine Benchmarks, FRAPS, GFXBench 3.0 (mobile)

# Other IDEAS

- Android/iOS/WP10 phone benchmarking (like **Antutu**, **Geekbench**, **GFX**)
- Browser benchmarking (HTML5, WebGL, like **Basemark**)
- Operating system benchmarking (like **HBench**-OS)

## Any programming language

## Standalone and portable

Windows 8+ OS mandatory, other OS optional

## Easy to run

Click, double click, command line

No other input parameters

## Provide the code

**You will send me the source code**

**(you'll receive a bonus if you use Git)**

## Idea

Originality or innovation of an existing idea

## Application portability

Self explanatory

## Software design

Code organization, algorithm correctness, efficient and unbiased implementation

## Output presentation

User experience, ease of use, UI

# THANK YOU FOR YOUR ATTENDANCE!

# SEE YOU NEXT TIME!