# Computer Architecture



*Here you are and you have already successfully passed several courses at SoftUni, congratulations. But have you ever wondered about how exactly the hardware of a computer is designed? This problem description will give you a peek into the architecture of a computer system.*

## 1. Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

## 2. Problem description

Your task is to create a computer repository that stores CPU components by creating the classes described below.

**CPU (Central Processing Unit)**

You are given a class **CPU,** create the following properties:

* **Brand - string**
* **Cores - int**
* **Frequency - double**

The class **constructor** should receive **brand, cores, and frequency**.

Override the **ToString()** method in the following format:

**"{brand} CPU:**

**Cores: {cores}**

**Frequency: {frequency} GHz"**

**Note: Format the Frequency to the first digit after the decimal point!**

**Computer**

**Next**, you are given a class **Computer** that has **a Multiprocessor** (a collection that stores **CPU** entities). All entities inside the collection have the **same fields**. Every **Computer** will have **Capacity** of the motherboard, and **adding new CPU will be limited** by the Capacity. Also, the **Computer** class should have the following **properties**:

* **Model - string**
* **Capacity – int**
* **Multiprocessor – List<CPU>**

The class **constructor** should receive the **model** and **capacity**, also it should initialize the **multiprocessor** with a new instance of the collection.

Implement the following features:

* Getter Count - **returns** the **number** of CPUs
* Method Add(CPU cpu) - **adds** an **entity** to the multiprocessor **if** **there** **is** **room** for it. If there is no room for another CPU, skip the command
* Method Remove(string brand) - removes a CPU by a **given brand. I**f such **exists**, returns **true**, otherwise, returns **false**.
* Method MostPowerful() - returns **the most powerful** CPU(the CPU with the **highest frequency**)
* Method **GetCPU(string brand)** – returns the CPU with the **given brand**. If there is no CPU, meeting the requirements, return **null**
* Method **Report()** - **returns** a **String** in the following **format:**
  + **"CPUs in the Computer {model}:  
    {CPU1}  
    {CPU2}  
    (…)**"

## Constraints

* The **models** of the computers will be **always unique**.
* The **capacity** of the computer will always be with **positive values**.
* The **brand** of the CPUs will be **always unique**.
* The **cores** of the CPUs will always be with **positive values**.
* The **frequency** of the CPUs will always be with **positive values**.
* You will always have a CPUs added before receiving methods manipulating the Computer's multiprocessor.

## Examples

This is an example of how the **Computer** class is **intended to be used**.

|  |
| --- |
| Sample code usage |
| *// Initialize the repository* Computer computer = **new** Computer(**"Gaming Serioux"**, 4);  *// Initialize entity* CPU cpu = **new** CPU(**"AMD Ryzen 5"**, 6, 3.7);  *// Print CPU* Console.WriteLine(cpu);  // AMD Ryzen 5 CPU:  // Cores: 6  // Frequency: 3.7 GHz  *// Add CPU* computer.Add(cpu);  *// Remove CPU* Console.WriteLine(computer.Remove(**"Intel Core i5"**));  // False  CPU secondCPU = **new** CPU(**"Intel Core i7"**, 8, 4);  CPU thirdCPU = **new** CPU(**"Intel Core i5"**, 8, 3.9);  *// Add CPU* computer.Add(secondCPU);  computer.Add(thirdCPU);  CPU mostPowerful = computer.MostPowerful();  Console.WriteLine (mostPowerful);  // Intel Core i7 CPU:  // Cores: 8  // Frequency: 4.0 GHz  CPU receivedCPU = computer.GetCPU(**"Intel Core i5"**);  Console.WriteLine(receivedCPU);  // Intel Core i5 CPU:  // Cores: 8  // Frequency: 3.9 GHz  Console.WriteLine(computer.Count);  // 3  Console.WriteLine(computer.Remove(**"Intel Core i5"**));  // True  Console.WriteLine(computer.Report()); // CPUs in the Computer Gaming Serioux:  // AMD Ryzen 5 CPU:  // Cores: 6  // Frequency: 3.7 GHz // Intel Core i7 CPU:  // Cores: 8  // Frequency: 4.0 GHz |

## Submission

Zip all the files in the project folder except **bin** and **obj** folders.