Hello everyone,

My name is Radu I am the Bootcamp coordinator. Welcome!

Bootcamp – is the first learning you do here with us and it’s a quite unique experience at City, but even beyond, when compared to other Universities.

**Bootcamp is an intense programming experience.** Over two weeks you will spend most of your days learning about and applying programming concepts. I’ll go more into the schedule of Bootcamp and discuss some practicalities in a few moments but a few words first.

We start your UG degree with Bootcamp because knowing how to program **is essential to being a computer scientist**. Also, Most of the modules you will take at City will involve some programming. We really want all our graduates to be competent programmers.

We recognize that there probably is quite a spread of **exposure to programming** in this room. For some of you Bootcamp will be the first contact with progrmmaing. If so, don't worry, we'll start from the basics and work up from there. On the other hand, we usually have a few quite competent programmers joining as well. If you are one of those, then no worries, bootcamp provides many opportunities for you both to shine and learn something new.

Given this range of skills among you it’s really important to keep in mind that you should not compare yourselves to others, especially if you haven’t programmed before. It's about where you are today and where you end up in two weeks, about how much progress \*you\* make, \*individually\*. We are quite confident that by the end of Bootcamp, you'll be able to write pretty neat programs that you can be proud of. They might not be as flashy or complex as other students' because you all start from relatively different levels here...

Aaand.. we want you to be resilient. And be confident that each of **you \*can\* learn how** to program. I'm saying this because it’s important that you have the right expectations. It’s impossible for someone to master programming in two weeks and we don’t expect that from you. If anything, we expect that somewhere during those two weeks you’ll hit a limit where you feel that you brain just doesn’t want to absorbe more, that you cannot keep up with the content. It might be this Thursday, or Friday, i or sometime next week. We expect it because we’ve seen this in the past few years in most of our students.

If at the end of the two weeks Bootcamp you will have felt a bit overwhelmed, don’t worry. After Bootcamp you’ll have a full first year to experience and **learn Java at a much slower pace**. Bootcamp is just a crash course taster – we know we throw a lot at you – and we just want you to do your best. Bootcamp is not assessed, you won’t get a mark so don’t stress too much.

But … do challenge yourself and especially **put the time in** – spend many hours a day programming. Programming is a skill, just like riding a bicycle or learning a foreign language. You don't learn by ... reading information before an exam, you learn by continuously and persistently doing and doing and doing. In the beginning sometimes you will feel like you are not making progress, like programming is not for you. But that’s how it is with any new skill and with time it will get better. The only important question to ask youself is “Have I spent the needed 10-20 hours this week on my programming?” If the answer is yes, then don’t worry. You will become a competent programmer.

Beyond programming, Bootcamp is also like an extended induction, a way to introduce you to University learning and to City, in particular.

First, quite a few of my colleagues are involved in Bootcamp and you’ll meet them. Here in Bootcamp they will deliver lectures and help with practicals. They will become familiar faces and that’s nice cause you’ll meet them again in coming years in other Modules. You’ll see that they are all nice and approachable and here to help you. And of course you’re going to meet your fellow students and make friends.

Second, you’ll experience the campus and the way we deliver our teaching. For example City is a bit of labyrinth and in Bootcamp you’ll do a lot of walking around and finding your way to rooms. That will be useful!

You’ll also find out how to access and engage with our online resources such as Moodle, online meetings, who to ask or email when you run into trouble and so on.

So, to recap: Bootcamp is an intense programming experience. We want you to spend lots of time programming, and challenging yourself, and learnin as much as you can. But we also want you to have the right expectations – nobody becomes a star programmer in two weeks – and don’t compare yourself to other. And also… use Bootcamp to get to know us and settle into your new student life.

**Now for some practicalities:**

After a year of online teaching we are back on campus. City is committed to delivering most of our teaching in person. This means that most lectures, labs, and activites in general will be on campus and that you should plan to come to campus if you want to get the fullest learning experience.

This being said there will be exceptions. For example, we have colleagues who are vulnerable or live with people who are and we must protect them by allowing them to work from home. We also learned from our online experience last year that some activities work well online, sometimes even better than in person. We don’t want to throw this lesson away. So - there will be some lectures, labs, or activities that will be online.

We also know that some of you or your colleagues who are not in the room today might not be able to come in person at some point – you fall ill, you need to self isolate, or you run into travel restrictions. So, using some of our learnings from last year, we are enhancing the online support that we offer to help out students who are in such situations. But that being said … we cannot fully double-deliver in-person and online; that is not all in-person learning experiences will be mirrored online or have the same level of support online. Attend in-person for the full learning experience, online is mostly just for getting-by if need be.

This is a general plan of how we want to teach you this year and Bootcamp reflects this plan. It combines in-person and online learning, but with a focus on in-person presence.

Last week during induction you were also shown a few slides with guidance from the University on Covid-related safety measures. These are important – please work with us as a team to create a safe environment. Wear masks, use lateral flow testing as much as possible, and get vaccinated. Bootcamp is an ideal place for a virus to spread – we are all together for many hours a day. We know how in previous editions by the end of Bootcamp colds were running amok. Let’s increase our chances of actually having a second week of Bootcamp and a term after that and all be a little safety aware.

All right - Moodle is the official go-to place for everything. Lecture materials, online meetings, exercises, explanatory videos… everything is there. Or rather will be there since we are releasing content gradually as the two weeks unfold. For now you see materials for today, tomorrow morning we’ll make visible the ones for tomorrow, and so on.

At the top of the Moodle page you’ll see the Bootcamp schedule: Bootcamp lasts two weeks. On most days we have two sessions of 3h – a morning one and an afternoon one. The first hour is a lecture in this format, with everyone together and one of the lecturers leading it. The next two hours are a practical or a lab. You’ll be in a smaller room with about another 35 colleagus and two of our staff who will help you out apply the new concepts you learn. Exceptions from this schedule are Weds, when we have just one session in the morning, and the last Friday when we have a Showcase – more about this in a second.

Mondays, Tuesdays, and Thursdays happen on campus. You are expected to be here in person to attend the lecture and then go to the lab. If you cannot make it though – for legitimate reasons - then there is this meeting called – standed-at-home which will provide an online version of the practicals. However - this is important - the lecture will not be live streamed there. You will be able to read through the content of the lecture and watch a recording of the lecture later but there won’t be a live online lecture during these days. Yes? So – if you can’t make it, read through the lecture materials on your own during the time allotted for the lecture then join stranded-at-home for practical help.

Weds and Fridays will happen online. You follow this link to join the online lecture or activity. Then, at the end of the lecture you quit that meeting and switch to your ‘online lab’ meeting. Note that during these days the stranded-at-home meeting doesn’t exist – because essentially we are all staying at home and there are dedicated online lectures and labs.

**The showcase** - will be an opportunity for you to reflect about how much you’ve learned during Bootcamp and show some of that to others. It will also be an opportunity to create and add **the first piece of work to your programmer’s portfolio**, one that you will develop while at City.

Concretely, for the showcase, we will ask you to work on a small, often very small, project that uses and showcases the new things \*you\* learned in those two week.. and then share that work with others. We’ll describe how this will work exactly on Friday but what you will want to know now is that this is not something that will be assessed. No marks, no stress … our hope is for you to challenge yourself and have fun with it.

So.. any questions on this introductory part?

So, lets start this first lecture by geting to know each other.

I’m a lecturer at City. I’m coordinating this Bootcamp but you’ll meet me again next term when I’m teaching the second part of Java programming – at that point you’ll already be much better at programming and you’ll get even better becasuse I’ll have you build a nice, functional 2D computer game.

Also like all your lecturers here at City, I’m not just a lecturer/a teacher… I’m also a researcher. I’m interested in data visualization and interaction – Data vis is about how we can visualize large and complex data in ways that allow us to find interesting stuff, insights, in the data. A chart or a graph are very simple but already quite useful examples of data visualizations but as a data visualization researcher I’m dealing with more complex data such as brain models or genomic data. The good thing for you is that my research is often connected to my teaching, I take things that I learn from my research and pass them on to you – For example, these screenshots are from applicaions I wrote, both using Java. And Java is precisely what I will be teaching you!

And of course I have a life out of Uni. I’m Romanian, have done my PhD and worked for a bit in the US, before moving back to Europe in 2016. This is Julia and she takes a big chunk of my life since about a year and a half ago. Lots of fun – but also lots of noise … and poo. Very likely you’ll get to hear her in one of the online meetings.

Julia also threw a wrench into my main hobby … wichh used to be dancing and teaching Argentine tango late into the night… that’s a bit hard to do with a toddler. But she certainly came at right time since lats year it was impossible to dance anyway.

Aaand, I also like to move .. – cycling, running, hiking, skiing … they are are all good fun for me.

That’s me. And I’m also approachable and here to help you.

So, lecture 1. I’ll introduce Processing, a programming environment for creating graphical and interactive applications. Processing build on or extends Java, one of the most popular programming languages. In essence you’ll learn Java but in an environment that makes it fun to do so.

By the end of the session you’ll learn: 1,2,3

All lectures contain a mixture of information, theory, and practical tasks. The tasks are probably the most important bit since as I’ve mentioned learning programming is all about doing and practicing.

You notice that the lecture materials are presented as webpages and you’ll also notice that these pages have a consistent visually layout.

Coding: have a go at trying something out rather than just reading about it.

Core tasks: fundamental tasks that teach and test what we’ve discussed in lectures. You need to do all of them. During labs prioritize them. If by the end of a lab you still have any left, do them as homework.

Aside

Tips

Embedded sketches on the right

Copying code: how you can do it; this way you can easily try things out but it’s essential that you write code yourself so outside lectures we’ll ask you not to do it.

Do you have processing installed?

Sketch and your own sketch book.

Draw a rect.

***A computer program is a set of precise instructions that gets the machine to do stuff***. In Java they are ordered.

Point out punctuation. ***The code you write must be free of syntax errors before the computer can understand and run it.***

***The shape of a command. Commands are ordered.***

***Coordinate system***

***size command; coordinates=pairs of numbers rep distances along axes; origin is at top left***

**A location of something is specified with two numbers in the order (x,y) that indicate the distance from the top-left corner in the horizontal and vertical directions respectively.** For something to be visible

Other shapes: lines, triangles, irregular quadrilateral

Each pair of numbers in a drawing command relates to a point or a size in the coordinate system. To see how they relate to the shape of a sketch, check reference.

Drawing styles:

* Dull and gray but we can style: fill, stroke, linewidth. They don’t produce and immediate visual effect. “sets the style of any drawing done in your sketch *after the style has been set*. “
* Stroke and fill take colors. Color are specified as triplets.
* Background command
* strokWidth

Comments///!!

Animation

– the idea of method (group of commands that do something);

* two methods that are built in: setup and draw
* how a method looks like (void – ignored, the name of the method; no semicol; curly brackets called braces; indent your code)
* but why write our code like this? Important behavior of Processing.
* It’s difficult to notice that the sketch redraws 60x sec because the rect is always drawn at 40 40.