Chapter 4
A Survey on Computer Programming Learning Environments

Ricardo Alexandre Peixoto de Queirós
Polytechnic Institute of Porto, Portugal

ABSTRACT

We are assisting the rise of online coding environments as a strategy to promote youth tech employment. With the growing importance of the technology sector, these type of technical training programs give learners emergent tech skills with a big impact and relevance to the current professional market needs. In this realm, MOOCs (massive open online courses) and online coding bootcamps are two increasingly popular options for learners to improve their code development skills and find work within a relatively short amount of time. Among all the features available on these environments, one stands out, which is the code generation. This chapter aims to detail and compare the most popular solutions for both learning contexts based on several criteria such as impact and maturity, user groups, and tools and features. In the features field, the authors highlight the code generation feature as an efficient way to enhance exercise resolution.

INTRODUCTION

Learning computer programming can be a lonely, complex, and demotivating process (Ala-Mutka, 2007), (O’Kelly & Gibson, 2006), (Robins & Rountree, 2003). These issues have been addressed in the last years, with the appearance of several on-line learning environments trying to leverage coding education and make it accessible to
everyone, even those with absolutely no coding experience or knowledge (Verdú et al., 2012), (Xavier & Coelho, 2011). These environments come in various formats ranging from non-interactive approaches (e.g. YouTube channels, blogs, books) to integrated and interactive solutions (e.g. intelligent tutors, on-line coding providers).

Nowadays there is an enormous demand on the technology sector to be up to date with the latest frameworks and languages. Regardless whether you are a coding newbie or a mature developer, you have several options, besides a computer science degree, to improve your programming skills. In this realm, MOOCs (Massive Open Online Courses) and Online Coding Bootcamps are two increasingly popular options for learners to improve their development skills and find work within a relatively short amount of time. While these two are excellent alternative learning contexts, the two options still have very distinct differences (Church, 2016).

A MOOC is an online course, usually available without charge, where learners can choose their own learning pace and direction. MOOCs are free educational courses often delivered by renowned university professors that typically feature a mix of downloadable readings, quizzes, discussion boards, video content and peer-to-peer assessment. The goal of MOOCs is to reach a much larger audience than traditional courses can accommodate. Often, MOOCs offer certificates for a fee which are awarded on successful completion of a course, and transferable college credit.

An Online Coding Bootcamp, on the other hand, is an intensive and paid course, usually eight to twelve weeks in duration, which offers hands-on training, career guidance and job assistance. These types of platforms involve a greater time commitment for the learner and are more suitable for who wants to quickly master a specific language (or stack) and get a technical job.

Both type of environments offers several features to foster code practice. One of the most important is the code generation feature. In fact, several environments deliver skeleton code that the students should complete to meet the problem requirements. Other feature typically used is the delivery of buggy programs. In this case the students would have to find logic errors in the program thus stimulating valences as debugging and testing. The rationale is simple: with the delivery of skeleton or buggy programs, the “problem-solving” issue is softened and the students’ working memory is free to build a new mental model of the problem to solve.

In this paper we will focus our attention on both learning contexts. In Section 2 we review the existent learning environments. In Section 3 we focus on the code generation techniques of the learning environments. Finally, we summarize the contributions of this research.
RESTful Web Services Development With a Model-Driven Engineering Approach
[www.igi-global.com/chapter/restful-web-services-development-with-a-model-driven-engineering-approach/219984?camid=4v1a](www.igi-global.com/chapter/restful-web-services-development-with-a-model-driven-engineering-approach/219984?camid=4v1a)

Operator Overloading as a DSL Parsing Mechanism
[www.igi-global.com/chapter/operator-overloading-as-a-dsl-parsing-mechanism/219983?camid=4v1a](www.igi-global.com/chapter/operator-overloading-as-a-dsl-parsing-mechanism/219983?camid=4v1a)