Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Arduino and a coffee machine.

Orfanakis V. ¹, Papadakis St.²

¹1st Vocational School of Agios Nikolaos, Crete, Greece
vorfan@gmail.com
²Department of Preschool Education, University of Crete, Crete, Greece
stpapadakis@gmail.com

Abstract

In this paper, we are going to present an approach for teaching fundamental programming concepts to novice programmers. Our proposed methodology seeks to take advantage of multiple technologies, such as the Twitter, the Python programming language and an Arduino board. The purpose of this approach is to give students a chance to: a) learn fundamental programing concepts, b) realize the development in the field of programming languages and c) realize the communication capabilities between seemingly unrelated devices. The novelty of the present approach is that the described activities: (a) promote students' interests and b) were based on the use of open source software.

Keywords: Arduino, Python, Twitter, Novice programmers, Basic programming concepts

1. Introduction

The teaching of algorithmic problem solving and fundamental programming concepts on novice programmers is considered one of the great challenges in computer science (Robins, Rountree & Rountree, 2003). Several studies indicate that programming is, for most students, an unattractive activity (Saeli, Perrenet, Jochems & Zwaneveld, 2011) and often novice programmers hold misunderstanding and misconceptions (Papadakis, Kalogiannakis, Orfanakis & Zaranis, 2015). Since the early 1970s, a wide range of approaches have been proposed to improve novices’ learning of programming (Robins et al., 2003; Kelleher & Pausch, 2005). Those approaches focus on the heart of the matter: how to teach novices to construct valid programs while at the same time to trigger their interest in programming and exempt them from the mental workload of a strict syntax memorization (Papadakis et al., 2015). An effective
approach is the teaching of basic programming concepts through cooperative learning activities based on the principles of constructivism and in the use of examples relevant to students' everyday experiences (Papadakis & Orfanakis, 2014).

2. **Students difficulties in learning programming**

Why we teach our students programming? Because programming is a much valuable skill as it promotes creativity, problem-solving and project-design skills, such as reasoning logically and contributes to the development of problem solving and computational thinking (Brennan, 2009; Berry, 2013). It is generally accepted that the teaching and learning of programming is normally characterized by some difficulties, which mainly occur in the construction of an algorithm or program (Papadakis, Kalogiannakis, Zaranis & Orfanakis, 2016; Tollervey, 2015). According to the classical teaching approach, students are taught a general purpose language (Pascal, Basic, C, Java, etc.). This method of teaching is problematic, as those languages include multiple commands which in combination with the strict structural details constitute a large amount of information that must be mastered by the students, forcing them often, to be more concerned with the language technical details and not focused on fundamental concepts and programming techniques. Papadakis et al. (2017) suggest the combined use of appropriate activities and solving selected problems in a computer lab using real programming environments. A constructive approach to learning programming requires appropriate teaching environments that on the one hand help students to solve problems, and effectively address the aforementioned misconceptions and difficulties on the other (Zaranis, Kalogiannakis & Papadakis, 2013; Papadakis & Orfanakis, 2014). This approach emphasizes on a pedagogical design for teaching novice programmers as the emphasis has shifted from teaching a strict language syntax to the development of critical and analytical thinking through problem solving (Papadakis, Kalogiannakis & Zaranis, 2016).

3. **A brief description of the hardware, software and services which are used in the teaching intervention**

3.1 **The Python programming language**

Python is one of the most widely adopted general purpose high-level programming language (Summerfield, 2010). However, Python has some powerful features that make it an ideal language for novice programmers (Tollervey, 2015): a) Ease of learning, b) Existence of an interactive shell which
allow programs to execute in an easy and fast way, c) Existence of various libraries which facilitate the learning process, such as turtle and TkInter library, d) Portability, and e) Installation flexibility.

3.2 The Arduino platform

Arduino is an open-source electronics platform based on easy-to-use hardware and software (Arduino, 2016). The Arduino platform bridges the computing world with the physical world, allowing the user to connect simple sensors and output devices to a computer (Desai, 2015). The characteristics that have made Arduino platform ideal for educational use are the following: a) it is reasonably priced, b) it is easy to use and programming, c) it includes an open-ended and expandable software and hardware platform, and d) despite its short existence, it has already had a large number of integrated projects and libraries.

3.3 The Twitter service

Twitter is a Web 2.0 service. It is the second most popular social network after facebook. Twitter as a short message communication tool allows someone to send out messages (tweets) up to 140 characters long to people who follow him (followers). Twitter provides several ways of connection. Any registered user of Twitter can send a Tweet via the official Twitter app, or 3rd party tools using computers or smart mobile devices.

4. The teaching intervention

The teaching of programming at school can be performed under various instructional strategies, teaching styles and programming environments (Kordaki, 2012). In the teaching intervention we designed, our main objective was students to learn basic programming concepts during open-inquiry and structured laboratory activities in an easy and attractive way close to their interests. An attempt was also made during the intervention of the six cognitive objectives of Bloom’s Revised Taxonomy Model to be achieved (Remembering, Understanding, Applying, Analysing, Evaluating, Creating). Additionally, we considered the tools which were used to be modern, with low costs, to meet students needs, and if it was possible to be able to be reused outside the the formal classroom context.

For this reason as a teaching intervention, was chosen a project in which all the four tools were combined: the Arduino platform, the Twitter, the Python programming language and a coffee machine. Bear in mind the "smart homes" and the "Internet of Things", our proposal also constitutes students’ first contact
with the latest technology concepts. The costs of implementing this technology-based intervention was 25 euros and included the purchase of a relay and the Arduino board. The central idea of the intervention is as follows. Depending upon their number, students are divided into functional groups of 3-4 persons. Subsequently, they will create a new Twitter account, which will be used to send a text message (tweet) with a special content e.g. "Start" to a remote computer. In the remote computer, a special application will receive that text message (tweet). The students will create that application using the Python programming language. This application, in turn, will send an appropriate message to the Arduino platform (via USB port). Subsequently, Arduino platform will trigger a relay which in turn will turn on a coffee machine. The same procedure shall be followed for turning off the coffee machine. The students will send a text message (tweet) with the special content "Stop". Figure 1 shows the project workflow.

![Project workflow](image)

**Figure 1.** Project workflow

### 4.1 The Twitter phase

In the twitter phase, each group of students will be responsible for implementing communication between the Twitter service and the receiver application (which will be developed in a later stage in the python language). For this to happen, each team will undertake to explore the meaning and usefulness of the Application Programming Interface keys (API keys). In this phase it is very important for the students to realize that in order to be able two different applications to communicate with each other without using a user name and a password each time, it is necessary the use of API keys and that each modern application such as Facebook, Twitter, etc. provides users with such keys. Figure 2 illustrates the API key of a twitter account.
4.2 The Python phase

In the Python phase, each team of students has to develop an application using the Python language. The application will act as a tweet receiver. If the message (tweet) contain the word "Start", the application will send a message through USB port which will trigger an application in the Arduino platform. If the message contains the word "Stop", respectively the coffee machine will be switched off (Figure 3).

```python
## Welcome message
print('Ευχαριστώ στην Απομακρυσμένη Διαχείριση!')

def drietwit():
    status = []
    x = 0

    status = api.GetUserTimeline('Remote_Tweet') ##grab latest statuses
    checkIt = [s.text for s in status] ##put status in an array
    drip = checkIt[0].split() ##split first tweet into words

    ## check for match and write to serial if match
    if drip[0] == 'start':
        print('To tweet ελήφθη. Ανάβει την καφετιέρα'
          ser.write('1')
    elif drip[0] == 'stop':##break if done
        ser.write('0')
        print('Το tweet ελήφθη. Σβήνει την καφετιέρα και περιμένει.'
    else:
        ser.write('0')
        print('Περιμένοντας ένα tweet')
```

Figure 3. Code snippet in Python

4.3 The Arduino phase

In the Arduino phase, each team of students has to develop an application which when it receives an appropriate signal through the USB port (to which is connected) will activate a relay in which the coffee machine is connected. More specifically, the the pin in which the relay is connected, will "get" the value
HIGH or LOW depending on the signal received by the Arduino platform from the Python application (Figure 4).

```cpp
int relayPin = 12; // LED connected to digital pin 13
int incomingByte = 0; // declare incoming byte

void setup(){
  // initialize the digital pin as an output:
  pinMode(relayPin, OUTPUT);
  Serial.begin(9600); // set up Serial library at 19200 bps
  Serial.print("Arduino is ready!");
}

void loop(){
  if (Serial.available() > 0){
    // read the incoming byte:
    incomingByte = Serial.read();
    Serial.println(incomingByte);
    if (incomingByte == 49)
      digitalWrite(relayPin, HIGH);
    else
      digitalWrite(relayPin, LOW);

    // say what you get:
    Serial.println("I received: ");
    Serial.println(incomingByte, DEC);
  }
}
```

**Figure 4.** The code in the Arduino IDE and an indicative worksheet

### 4.4 The coffee-machine face

In the coffee-machine phase, each team of students firstly has to understand how a relay works and then to incorporate it into a power strip. The aim is the coffee-machine which is permanently connected to the power strip, to be activated or not via the integrated relay which is directly connected to the Arduino board. In this phase, the activity becomes interdisciplinary with the involvement of Natural Sciences as students is important to understand electricity basic principles, the way relays work etc (Figure 5).

**Figure 5.** A relay on a power strip
5. Summary and future work

In this paper we presented a didactic approach to help novice programmers to learn basic programming concepts in an easy, and attractive way close to students interests. The experimental implementation of the teaching approach took place in the school year 2014-2015, with the participation of five students as part of a school activity program, in a provincial high school in Crete, Greece. Although the results of case studies are difficult to generalize, the evaluation of the implementation through students semi-structured interviews showed encouraging results. Students through their involvement seem to have understood the basic concepts of programming and technology as they engaged in practical work in an interdisciplinary authentic environment. In our case, more analysis is required to understand the areas that require further development both from the conceptual and practical part; therefore, further case studies should be conducted involving a large number of students. Our future plans also include the enhancement of this teaching approach with the creation of a twitter application by the students using the programming environment App Inventor for Android in order to familiarize themselves with mobile devices programming.

Acknowledgement

The idea for this teaching approach was based in a post in the website Instructables named "Tweet-a-Pot: Twitter Enabled Coffee Pot".

References


