Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 176 (2015) 291 – 298

Procedia
Social and Behavioral Sciences

IETC 2014

# An application for fundamental computer programming learning

Ozan AKI*[a] Aydın GÜLLÜ[b] Erkan KAPLANOĞLU[c]

[a,b]*Trakya University Ipsala Vocational School, EDIRNE 22400, Turkey*
[c]*Marmara University Technology Faculty, ISTANBUL 34730 Turkey*

**Abstract**

Applied computer laboratory lessons could be unproductive because of many students in there. Correcting students' mistakes one by one is wasting lesson time. Especially for beginners, most of these mistakes caused by complex integrated development environments. In this study, we develop a client server application for computer laboratories. Developed application is able to compile programming language source code remotely. Thus, students don't need to make something out of the writing source code. Furthermore, instructors don't need to install compiler to the each computer in laboratory. For start lesson, it is enough that server has just been configured.



*Keywords:* computer programming education, computer laboratory, client server application

## 1. Introduction

Computers are used almost in all business areas. Therefore, it is inevitable that computer courses become parts of the school curriculum (Mayer, 2013). These courses are at the level of the computer operator for social classes while the technical classes need to be more specific and detailed. Most of these courses are consist of computer programming courses. Computer programming is a difficult and challenging subject area which places a heavy cognitive load on students (Mow, 2008).

It is obvious that programming courses done in the computer laboratory. Conventional way of teaching computer programming is installing compiler and integrated development environment (IDE) to all students' computers. While this is not a problem here for engineering students, but there are some difficulties for vocational school students

* Corresponding author. Tel.: +90-535-507-9611; fax: +90-284-616-3534.
*E-mail address:* ozanaki@trakya.edu.tr


doi:10.1016/j.sbspro.2015.01.474

because of they don't have any enough background knowledge about computer programming (Ismail, Ngah, & Umar, 2010).
One of the challenges faced by students is complexity of IDE. Spending time for configuring IDE for every new project and correcting issues caused by project creation mistakes in the limited time of lesson is wastes lesson time. By the number of computers in the lab this problem being more complicated.

In this study, we predict that an application for providing very simple interface for single file source code writing and simply compiling and running may help to students to concentrate writing code rather than configuring IDE. Therefore we developed an application using laboratory network infrastructure to achieve this goal. This application was designed for client / server structure. This design is consistent with laboratory network infrastructure. Teacher's computer implies the server while students' ones are clients. Client side of developed application does not need any installation; it is simply executable file like notepad application coming with Windows™ operating system. Server side of this application needs that preferred compiler tool path has been configured. Once server application is configured, it is ready to compiling. A typical laboratory network infrastructure is shown as Figure 1.

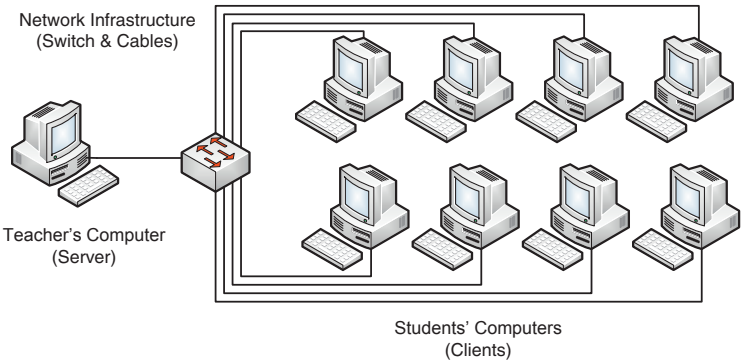



Figure 1 A Typical computer laboratory network infrastructure

## 2. Available programming environments

There are many programming languages available today. Many popular programming languages like C++, C#, Java, web scripting languages like PHP have same syntax notation with C for fundamental statements. Also almost all embedded chip manufacturers have C compilers for embedded development. Arduino is most popular example for this (Arduino, 2014). So we prefer C language for teaching to students. Therefore students can easily migrate own knowledge to many sections in programming like step up to advanced programming languages or drill down to embedded programming.

Every programming language needs a suitable compiler. Compiler translates the source code to the executable binary code for specific platform or processor. Compilers are an executable command line tool that takes source codes as input files and generates output files.

Source codes are simply text files. Writing source code does not need any specific tools. Basically a text file editor is enough for writing it. But many projects consist of multiple source codes that cause complicated compile sequence. For helping programmers to writing source code there are many advanced editors have been developed called as IDEs.

IDEs are designed for programmers and/or engineers that profession is computer programming. These IDEs contains advanced editor, compilers, tools and helpers. Table 1 shows available IDEs that suitable for using for education in laboratories.

IDEs showed in Table 1 are selected with these criteria: a) Designed for Windows platform. b) Supporting at least C / C++ language c) Having at least C / C++ compiler internally or externally. d) Usable for free without any license issue. These IDEs are tested on a single core CPU, 512MB RAM computer with Windows XP SP3 clean installation. This computer had a 16 Mbps internet connection. Previously installed components have been uninstalled and

temporary folders and registry have been cleaned before installation of every individual IDEs. Every installation wizard has been completed by its default options. Some setups contain its pre-requirement components for install these automatically while required. Some installations download its updated components from Internet while installing. After installation of every IDEs, a simple "Hello World" program has been compiled and ran for testing.

Table 1 Available integrated development environments that suitable for using in education

| IDE Name and Version | Pre-Requirements | Compiler | Setup Size | Installed Size | Setup Duration |
|---|---|---|---|---|---|
| Microsoft Visual C++ 2010 Express | - | Microsoft C++ | 175 MB | 385 MB | 16 min 30 sec |
| NetBeans 8.0 CPP IDE | JDK, MinGW | MinGW | 62 MB | 217 MB | 6 min 30 sec |
| Eclipse Luna IDE | JRE, MinGW | MinGW | 165 MB | 223 MB | 25 sec |
| Bloodshed Dev C++ 4.9.9.2 | - | MinGW | 9 MB | 60.7 MB | 1 min 50 sec |
| Orwell Dev C++ 5.7.1 | - | MinGW | 45 MB | 341 MB | 1 min 55 sec |
| CodeLite 6.1.1 | - | MinGW | 45 MB | 328 MB | 40 sec |
| C Free 5.0 | - | MinGW | 14 MB | 89.1 MB | 15 sec |
| CodeBlocks 13.12 | - | MinGW | 101 MB | 259 MB | 50 sec |

Bloodshed Dev C++ has small setup size and simple interface. Unfortunately Dev C++ development has been stopped in 2005 and does not support anymore. This IDE is still working on many systems but having some issues with newer operating systems (Bloodshed, 2014).

Orwell Dev C++ is based on Bloodshed Dev C++ and updated releases are available on its web site. Orwell Dev C++ has similar interface with Bloodshed Dev C++ (Orwell, 2014).

C Free is another small sized and simple interfaced IDE and does not need any pre-requirements (Arts, 2014).

NetBeans and Eclipse are based on Java and haven't got build in compiler. Java Runtime Environment (JRE) or Java Development Kit (JDK) must be installed before installation of these IDEs. Preferred compiler should install separately and these IDEs needs to have been configured before compiling (Eclipse, 2014; Netbeans, 2014).

CodeBlocks has similar interface to NetBeans and Eclipse. But setup file includes its pre-requirements and compiler (Codeblocks, 2014).

Microsoft Visual C++ 2010 Express is a part of Microsoft Visual Studio IDE. Supporting wide range of tools causes huge setup media and seriously long setup time (Microsoft, 2014).

CodeLite IDE was installed successfully but had crashed without any message. We could not able to run this IDE for presented test system (Codelite, 2014).

## 3. Application for teaching and learning fundamental programming

The common difficulty of these IDEs is installing, configuring and maintaining on many computers. Inspecting students' source code and correcting mistakes is another difficulty in laboratory. Finally, collecting students' work files is another problem in many times. Our developed application is designed to bring solutions to these problems. There are some studies that aim to similar purposes (Karkalas & Gutierrez-Santos, 2014; Rodrigues, Marques, & Martins, 2014).

Many computer laboratories have a local network that computers are able to communicate with others and able to connect to the Internet. There are many computer classroom management applications have been developed by thanks to network infrastructure. Developed application in this study is using network infrastructure in computer lab.

Application is designed in two separate modules called as client and server. Server module is running on teacher's computer. Client modules are running on students' computers. The preferred compiler should have been installed on teacher's computer. Client modules do not need any installation or configuration. Client module consists of one simple executable file. Once server module has ran, it broadcasts its own IP address to the all clients. Client modules captures these IP address and ask student to enter name for connect. Once after connecting to the server, it is ready to

compile and run. Server module lists connected clients by names and IP addresses. Teacher is able to see and any student's source code and console out by clicking on it.

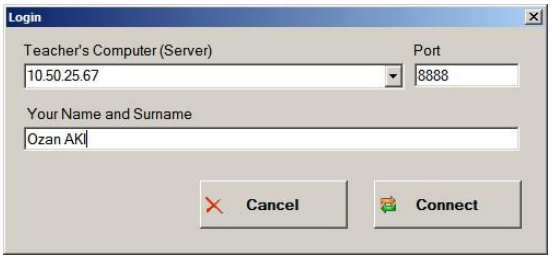



Figure 2 Client module's login screen

Client module interface is very simple like a notepad. There is a file menu that consists of file operations like new, open, save, close a file. Another menu has only one choice that named Compile & Run. When student click on compile and run menu, source code is sending to the server module for compiling. If code has any syntax error and compiling has failed, compiling results send back to the client module. These error and warning messages viewed on console window at the client application. Students can correct the code and try again until the compiling is successful.

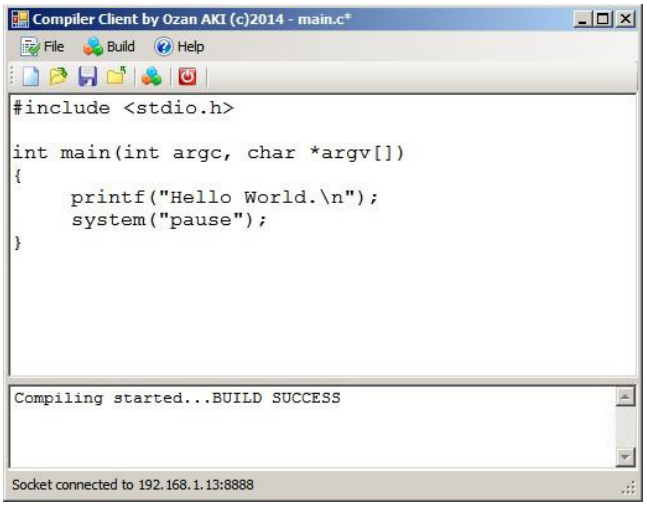



Figure 3 Client module's main screen

When any client code compiling is successful, compiled and linked executable file is send back to the client and automatically run that students can interact and view results. After the lesson or unit, teacher can collect all compiled source codes with students' names by one clicking into the any folder.

This application was developed by Microsoft Visual Studio C# Language with Dot Net Framework 4.5 for Windows platforms. Socket programming and threading techniques are used (Schildt, 2011). Transmission Control Protocol (TCP) is used for transmit source code and executable file. User Datagram Protocol (UDP) is used for dispatch server IP address. Every compiling process started as new thread for avoid blocks other compiling requests. Returned executable file is running as local console application in client computer. Students can fully interact with their built application.

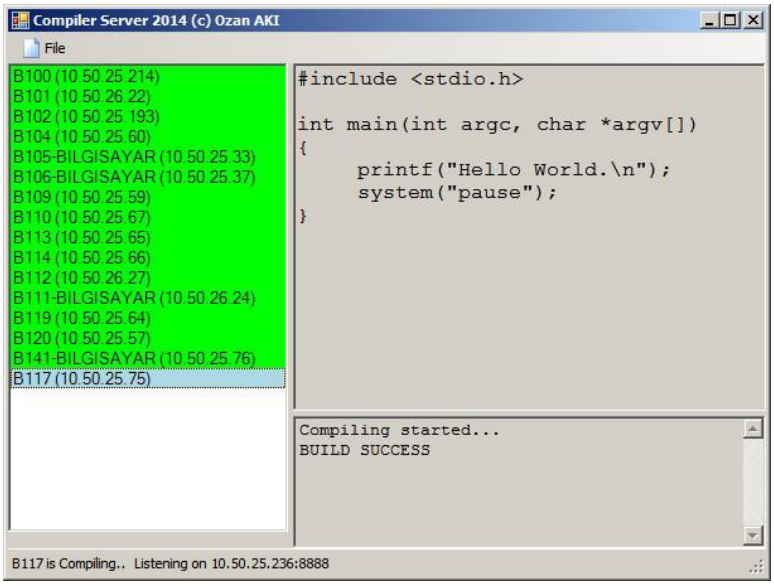



Figure 4 Server module's main screen

Advantages of this application can be listed as below;

- Does not need to install compiler to the all computers
- Teacher can see any students' source code and console results on their screen
- Teacher can collect all source codes by students' name by one click
- Students don't need to do any configuration. Students can just writing code and simply compile and run it.
- Client computers do not need to be a high-end hardware.

Probability disadvantages of this application are listed below;

- Clint module have not an advanced editor that helps and suggests while writing code
- Client module have not coloring source code key words
- Client module does not indicate an error by coloring line.

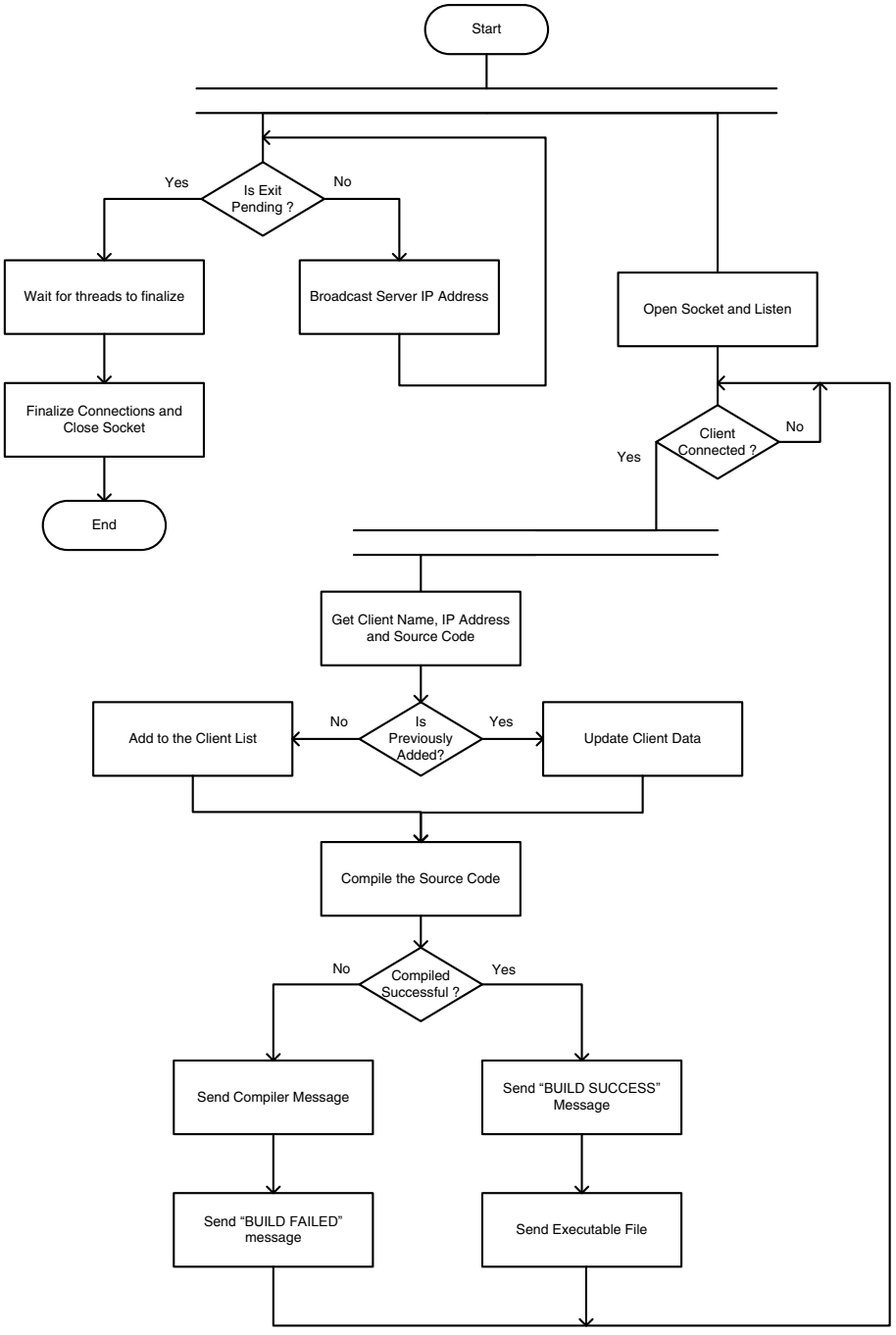
## 4. Discussion and results

Developed application is an initial attempt to realize the idea. Application is tested in computer laboratory with 16 clients. Minimalist GNU for Windows (MinGW) is used for compiler tool chain on server. All clients had compiled remotely its source code successfully. After all compiling processes, all students' source codes were collected easily by one click. Test results shows that this application could help teaching of computer programming but not tested on real lesson conditions yet.

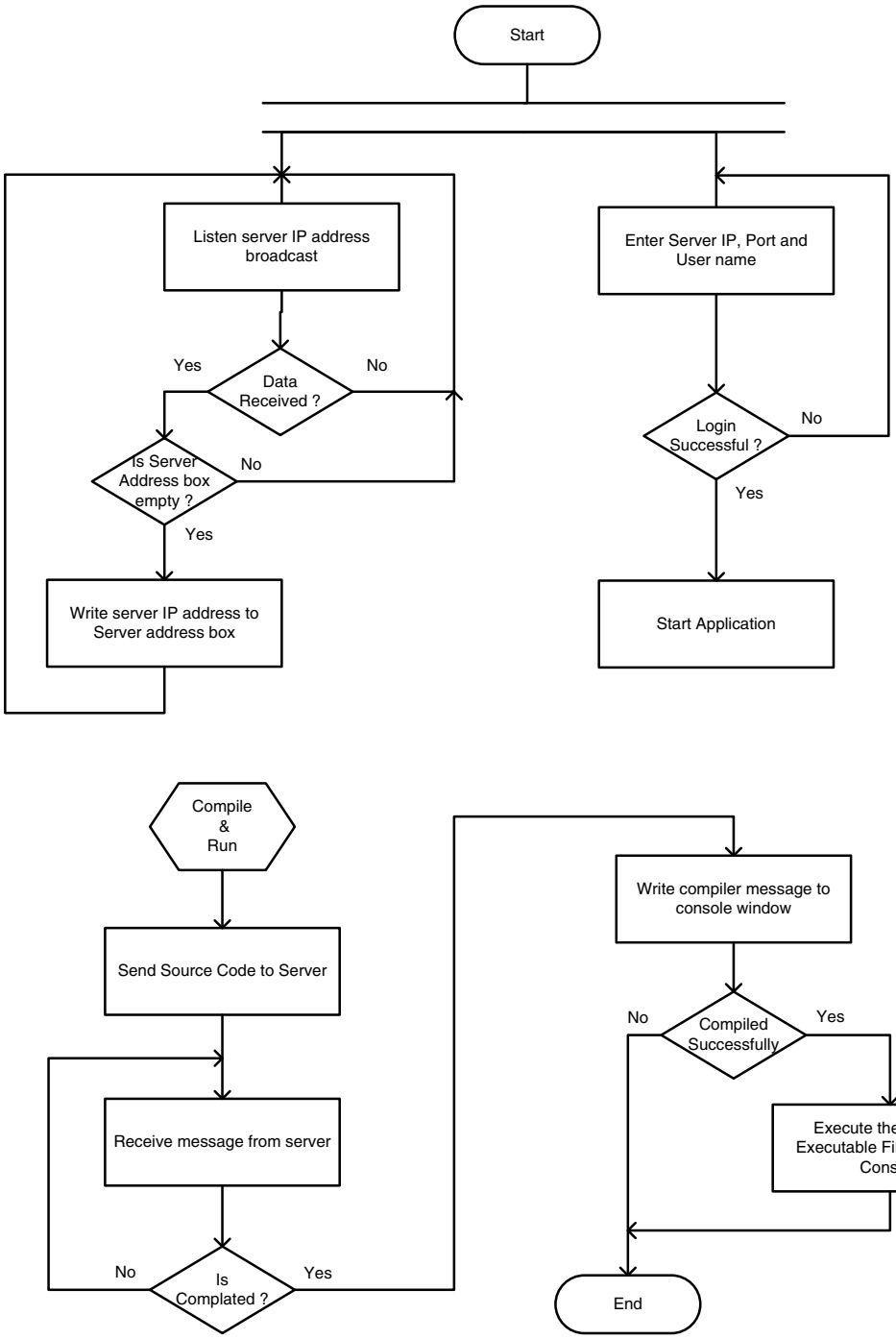
On the other hand, IDEs should not be considered to be disregarded by this application. This application is designed for beginner students in computer programming. After meeting students to the programming, of course students should learn using of least one IDE for programming.

## Appendix A. Flowcharts

### *A.1. Server Module Flowchart*



*A.2. Client Module Flowchart*



## References

Arduino. (2014). Arduino. from http://arduino.cc
Arts, P. (2014). C Free. 2014, from http://www.programarts.com/cfree_en/index.htm
Bloodshed. (2014). Bloodshed Dev C++. 2014, from http://www.bloodshed.net/devcpp.html
Codeblocks. (2014). Codeblocks. 2014, from http://www.codeblocks.org/
Codelite. (2014). Codelite. 2014, from http://codelite.org/
Eclipse. (2014). Eclipse Luna Ide. 2014, from https://www.eclipse.org/
Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional Strategy In The Teaching Of Computer Programming: A Need Assessment Analyses. *TOJET: The Turkish Online Journal of Educational Technology – April 2010, volume 9 Issue 2, 9*(2).
Karkalas, S., & Gutierrez-Santos, S. (2014). Eclipse Student (In) Activity Detection Tool *Open Learning and Teaching in Educational Communities* (pp. 572-573): Springer.
Mayer, R. E. (2013). *Teaching And Learning Computer Programming: Multiple Research Perspectives*: Routledge.
Microsoft. (2014). Visual Studio. 2014, from http://www.visualstudio.com/
Mow, I. T. C. (2008). Issues And Difficulties In Teaching Novice Computer Programming. In M. Iskander (Ed.), *Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education* (pp. 199-204): Springer Netherlands.
Netbeans. (2014). Netbeans Ide. 2014, from https://netbeans.org/
Orwell. (2014). Orwell Dev C++. 2014, from http://orwelldevcpp.blogspot.com.tr/
Rodrigues, M., Marques, F., & Martins, C. (2014). Teaching Advanced Computer Programming Topics Using An Automatic Validation Tool: Challenges And Solutions. *EDULEARN14 Proceedings*, 1541-1547.
Schildt, H. (2011). *C# 4.0 The Complete Reference*: ALFA.