

# Design and Development of C++ Quiz System for Android-based Platform

Jiping Li *School of Comput. & Inf. Sci.*

*Hubei Engineering University*

Xiaogan, China

oucljp2012@yahoo.com

Huaqing Xiao *School of Phys. & Electron. Inf. Eng.*

*Hubei Engineering University*

Xiaogan, China

xgxhq@126.com Hui Zhang *School of Econ. & Mgt.*

*Hubei Engineering University*

Xiaogan, China

zhang\_if@126.com

## Abstract

C++ is considered as one of the most popular programming languages and rated as the third most popular language next to Java. So far, most computer games, applications such as MS office suite, Adobe system, Google applications, Mozilla Firefox, thunderbird, Sybase, Oracle, MySQL, etc. are written in C++. However, some graduate's ability in C++ programming is poor and inefficient to meet the programming requirements for the talents of software service outsourcing. To this end, an Android-based C++ quiz system is designed and developed with Android Studio to check students' mastery degree of C++ programming and to improve their C++ programming ability. The Android-based C++ quiz system can randomly generate a testing paper, which is composed of choices, fill in the blanks, true or false, and fill in the program, by choosing different types of questions from the question bank. Testing results, which are obtained from both Android virtual device emulator and a physical device, show that the Android-based C++ quiz system is suitable for evaluating and strengthening the students' mastery of C++ programming.

## Index Terms

C++ programming, Quiz system, Android studio, SQLite

## I. INTRODUCTION

Since the first standardization of C++ happened in 1998, it has evolved into the latest revision, namely C++ 17, after major changes to the core language and standard library, and minor extensions. Now C++ is considered as one of the most popular languages and rated as the third most popular language next to Java and C [1]. Since C++ is a superset of C, all programs written in C will be able to run under C++ compiler. Both C and C++, which are developed by ATT Bell Lab, are used to write programs, including computer games, MS office Suite, Oracle and Sybase [2]. Moreover, top 10 best applications such as Adobe system, Google applications, Mozilla Firefox and Thunderbird, MySQL, etc. are written in C++ [3]. As a result, it is the first course for most of university students who major in computer, electronics, and electrical engineering and so on [4]. To speed up the development of software service outsourcing, more and more universities and colleges in China pay more attentions to the cultivation of talents with extensive programming skills. However, the graduates actual C++ programming ability is poor. There are several reasons such as the imperfections in teaching plan, lack of industry-experienced teachers for C++ programming. Nevertheless, the most important reason which cannot be ignored is that it is short of effective and convenient C++ quiz system to check and improve the students' ability in C++ programming. To remove this barrier, several web-based C++ learning or quiz systems are developed in the past year [5], [6]. However, the web-based C++ learning or quiz system depends greatly on the Internet. If there is no reliable network for web surfing, the C++ learning or quiz system cannot provide learning or testing service. In addition, micro-blog based platform for the teaching of C++ programming is also developed [7]. In [7], the authors present the introduction of a micro-blog platform into the classroom instruction of C programming at Xi'an Jiaotong University in China. The results of questionnaire survey of participants demonstrate that micro-blog plays a significant role in enhancing communication between teachers and students, thus, improving the acquisition of knowledge and learning interest. In [4], [8], the authors develop a novel integrated development environment (IDE) called Ubi-C for C programming based on Clang and LLVM. Using this IDE, users can directly write, compile, execute and debug their C programs on Android-based smart phones or tablets. However, the developed IDE only provides an environment for

This work was supported by the 2015 Key Research Subject of Hubei Provincial Education Science Planning under Grant No.2015GA038 and Teaching Research Project of Hubei Engineering College under Grant No.2015A37, and the 2015 Research Project of Provincial Teaching Reform in Hubei Universities under Grant No. 2015400.

users to write, compile, execute and debug C program, and has no function of checking how well the users understand and use the C/C++ to program. In addition, to use the online C++ quiz system, the users must have good network condition, with which they can access the Internet. Considering the traffic cost and inconvenience of using a computer anywhere and anytime, we are motivated to design and implement a C++ quiz system based on Android platforms such as smartphone, to test students' knowledge and focus on their weak points in C/C++ language learning. Our contributions are listed in the following.

- 1) We present the framework and workflow of the Android-based C++ quiz system, as well as related technologies.
- 2) We present core codes in detail to realize the related key functions of the Android-based C++ quiz system.
- 3) We present the testing results, which are obtained from both Android virtual device emulator and a physical device, to show that the Android-based C++ quiz system is suitable for evaluating and strengthening the students' mastery of C++ programming.
- 4) According to the feedbacks from students, teacher can constantly add different types of questions, in which the students are prone to make mistakes, to the question bank. Thus, the C++ quiz system can well test students' C/C++ knowledge.

The rest of the paper is organized as follows. Section II presents the research background and related technologies. Section III presents the design and development of the Android-based C++ quiz system. Section IV presents the testing results of the Android-based C++ quiz system in terms of Android virtual device emulator and a physical device. Finally, Section V gives the conclusions of this paper.

## II. RESEARCH BACKGROUND AND RELATED TECHNOLOGIES

In this section, research background and related technologies are presented to well indicate the motivation of developing the Android-based C++ quiz system and related developing technologies.

### A. Research Background

In recent years, software service outsourcing industry has developed sharply all over the world, especially in Asia. It is estimated that IT-outsourcing sector in India will be possibly rising to \$150 billion by 2020 [9]. As a main product of economic globalization, software service outsourcing plays an important role in promoting modern service industry development and adjusting the industrial structure in China. Now software service outsourcing in China is in a growing stage of industry life cycle [10]. It is reported in 2013 that the revenues of China's software industry is CYN 3.1 trillion, jumping by 24.6% year on year, which is much higher than the global average of 5.7% [11]. According to China's Ministry of Industry and Information Technology (MIIT), China's software industry consisting of 35,774 companies generated revenues totaling CNY 5.504 trillion ( US\$ 814.02 billion ) in 2017, growing 13.9% on year [12]. However, the sustainable and stable development of software service outsourcing needs programmers with rich experiences in C++ and Java programming, so how to improve students' C++ programming ability becomes very important.

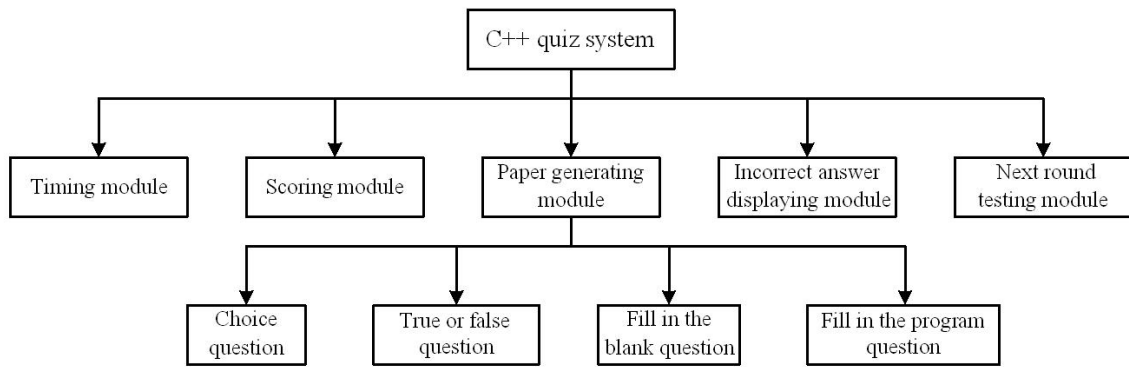
### B. Related Technologies

It is reported that 86% smartphones in the world run with the Android operation system, so the portable C++ quiz system is designed and developed for Android platform. To develop the Android-based C++ quiz system, the following developing technologies are involved.

- *Android Studio* is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.
- *Java Development Kit (JDK)* is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows [13]. It includes a private JVM and a few other resources to finish the development of a Java Application [14].
- *Android SDK* provides a simple Java API that enables your Android app to access a LoopBack server application. It enables you to interact with your models and data sources in a comfortable native manner instead of using clunky interfaces like AsyncHttpClient, JSONObject.
- *AVD Manager* provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator.
- *SQLite* is an embedded SQL database engine. Unlike other SQL database, the SQLite does not have a separate server process. In fact, the SQLite is a compact in-process library which implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

## III. SYSTEM DESIGN AND DEVELOPMENT

In this section, the composition of an Android-based C++ quiz system is firstly presented, then the function of each module is elaborated, and finally the development of the system is presented in detail.



**Figure 1.** The framework of the Android-based C++ quiz system.

### A. System Composition

The Android-based C++ quiz system is developed to check and improve the students C++ programming ability, to make the system more intuitive, the login process is neglected in our design. The students can directly browse the list of questions and answer them once the Android-based C++ quiz system runs. To well evaluate the students' programming ability, a timer is triggered when the Android-based C++ quiz system begins to work. The Android-based C++ quiz system is composed of paper generating module, scoring module, timing module, incorrect answer displaying module and next round testing module. The framework of the Android-based C++ quiz system is illustrated in Figure 1.

The paper generating module is used to generate C++ testing paper by randomly selecting a certain number of questions in the form of choice, true or false, fill in the blanks and fill in the program from pre-stored question bank. The scoring module is used to grade the learning effect of C++ programming language after the examinee's submission. The timing module works as timekeeper, which can tell the examinee the exact interval used in the quiz. The incorrect answer displaying module is used to feedback where the errors exist in the examinee's answer. The next round testing module is used to start the next new testing.

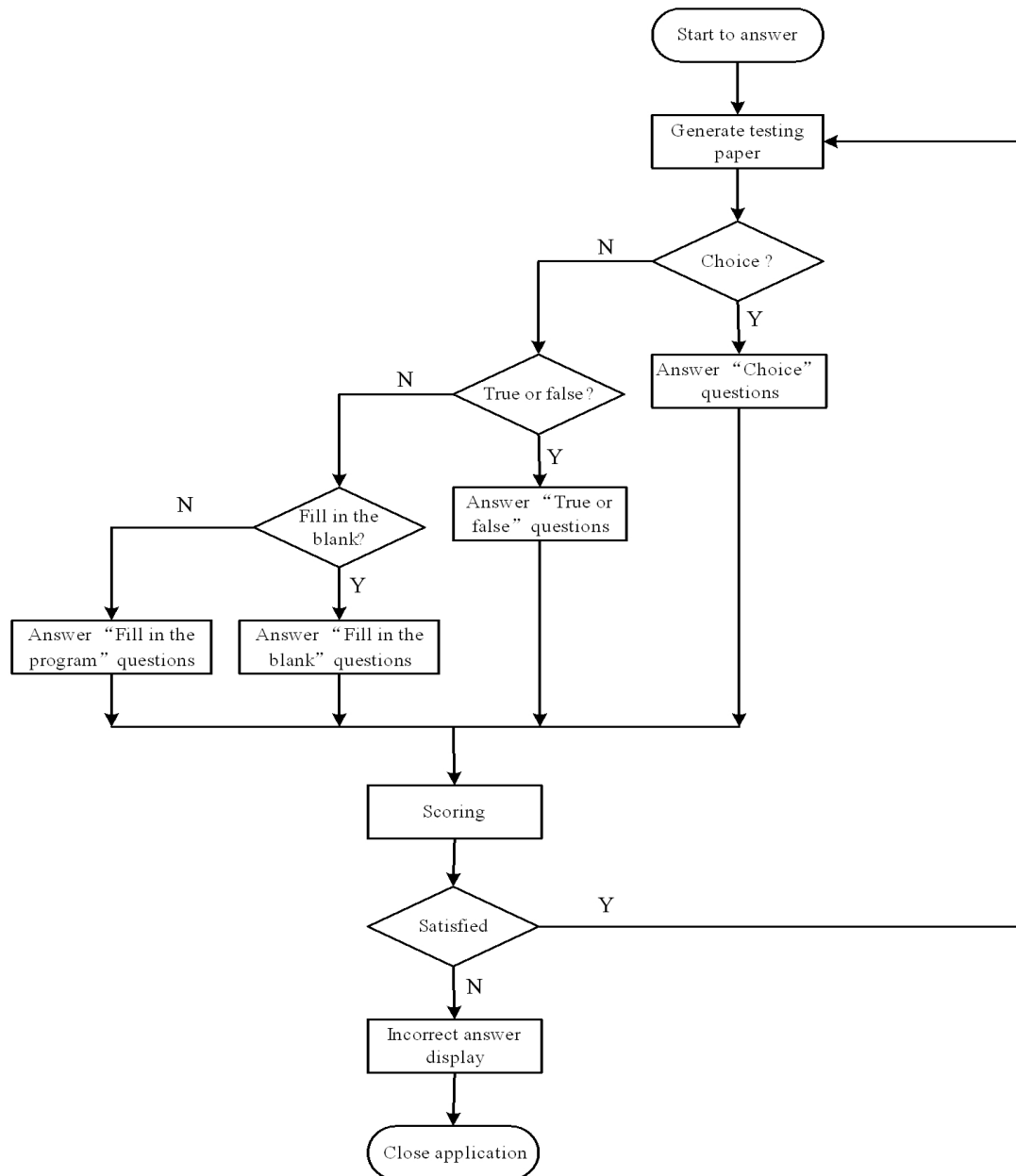
### B. System Function

The Android-based C++ quiz system consists of two pages (activities). The first one is the page used to answer questions, and the other is used to score the examinee's answers. When the Android-based C++ quiz system is launched, the system randomly selects 10 single-choice questions, 5 fill-in-the-blank questions, 4 true-or-false questions and 1 fill-in-the-program question to form a new testing paper, and then the system enters into the answering-question page, meanwhile, a timer is triggered to work. When the examinee finishes answering all the questions listed in the answering-question page, the system shifts to the scoring page after clicking the "submit" button. In the scoring page, the final scores of the testing are presented. If the examinee is not satisfied with the testing results, he/she can retrieve the records related to his/her question answering. Of course, if the examinee is satisfied with the testing results, he/she can launch the next round test by clicking the "Next testing" button. The work flow of the Android-based C++ quiz system is illustrated in Figure 2.

### C. Layout of the System

In Android-based application, layouts provide different levels of control over how visual user interface components are positioned and managed on the screen [15]. Several layout views such as constraint layout, linear layout, table layout, frame layout, relative layout, absolute layout, grid layout and coordinator layout, may be used within an Android user interface design. For some reasons, most Android application's contents do not fit the screen. For example, when answering the fill-in-the-program questions in each round of testing, the contents of the question are random and dynamic and it may grow beyond the screen size. If standard layouts such as *LinearLayout*, *RelativeLayout*, *FrameLayout* or *TableLayout* are used, the examinee won't be able to scroll and view the whole content of the question when the content grows, and data increases beyond the screen size. To avoid this weakness, a special kind of layout *ScrollView* is provided to let user view all the contents of the question. However, the *ScrollView* can contain only one control. If two or more controls are included in the *ScrollView*, error warning will generate, so a *RelativeLayout* and a *LinearLayout* is included in the *ScrollView*. The *RelativeLayout* contains an *ImageView* for timer, 3 *TextView*s for text display, a *RadioGroup* for choice, and 2 *EditText*s for filling in the blanks. The *LinearLayout* contains 3 *Button* controls, the first for backing to the last question, the second for going to the next question, and the third for paper submission, and one *Spinner* control for choosing different questions.

For the scoring page, *RelativeLayout* is adopted to arrange the controls in this page. There are 3 *TextView* controls and 2 *Button* controls in the *RelativeLayout*. The first *TextView* is used to display how much the examinee scores, the second *TextView*



**Figure 2.** The workflow of the Android-based C++ quiz system.

is used to inform the examinee how many questionshe/she answers correctly, and the third *TextView* is used to inform the examinee the number of incorrect answers. The first *Button* control is used to back to the question-answering page, and the second *Button*control is used to launch a new round test.

#### D. SQLite Database

In the development of Android-based app, a lightweight database SQLite is preferred instead of graphical database MySQL. The SQLite is an embedded database because it is provided in the form of a library that is linked in applications. It is a transportable, lightweight and free open-source and relational database system. In SQLite, the data is stored in tables, and you need to use SQL statements to create, read, update, and delete data.

- *Table Creation:* To randomly generate C++ testing paper, the system should select four types of questions from the question bank which consists of several tables. To this end, *SQLiteOpenHelper* class, database handler class (a subclass of *SQLiteOpenHelper*) are involved in theapp. The database is first initialized, and then the "test" table is created by using *onCreate()* method. The database initialization and "test" table creation are illustrated in Figure 3.
- *Insertion of Keywords and Data:* To implement the primary interface between the application code and underlying SQLite databases such as the ability to create, delete and perform SQL-based operations on databases, some key methods including

```

public class mydb extends SQLiteOpenHelper{
    private SQLiteDatabase db; //SQLite database
    public mydb(Context context, String name, CursorFactory factory, int version)
    {
        super(context,name, factory, version);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        //Create table for question bank
        db.execSQL("CREATE TABLE IF NOT EXISTS test" +
            " ( _id integer "+
            "primary key AUTOINCREMENT , timu VARCHAR," +
            "xuan1 VARCHAR,xuan2 VARCHAR,xuan3 VARCHAR,xuan4 VARCHAR,answer VARCHAR,cid integer)");
    }
}

```

Figure 3. The creation of “test” table.

```

//insert data indicating question type
db.execSQL("insert into choose values(1,'choice questions')");
db.execSQL("insert into choose values(2,'fill-in-the-blank questions')");
db.execSQL("insert into choose values(3,'true-or-false questions')");
db.execSQL("insert into choose values(4,'fill-in-the-program questions')");

```

Figure 4. Insert data in terms of question type.

*insert()*, *delete()*, *query()*, *execSQL()* and *rawQuery()* of *SQLiteDatabase* class are provided. In the Android-based C++ quiz system, *execSQL()* is used to fulfill the question bank. To distinguish different types of questions, choice question is labeled as field1, fill-in-the-blank question as field2, true-or-false question as field3 and fill-in-the-program question as field4. Insertion of Keywords and data is depicted in Figure 4 and Figure 5, respectively.

- *Data Query*: A virtual record set stored in the memory is obtained by using database query operation. If traversal of all records hopes to be performed, a pointer is needed to indicate the searched record. To this end, *Android.database.cursor* is used as an interface to provide this function. There are two query methods, namely *query()* and *rawQuery()*, in *SQLite*. The *query()* method is used to perform a specified database query and to return matching results via a cursor object. However, the *rawQuery()* is used to execute SQL query statement and to return matching results like *query()* method. The method *rawQuery()* requires just 2 parameters, with a single line “where” clause, while the method *query()* requires a few parameters with separate query at certain places, so the *rawQuery()* method is adopted in developing the Android-based C++ quiz system. The core codes for data query are listed in Figure 6.

#### E. Testing Paper Generation

In each testing, 20 questions including 10 choice questions, 5 fill-in-the-blank questions, 4 true-or-false questions and 1 fill-in-the-program question are randomly selected from the question bank to generate a testing paper. To this end, the method *nextInt(UP\_BOUND)* of class *random* is used to generate a random integer number ranged 0 ~ *UP\_BOUND* for randomly selected questions of different types. The core codes of testing paper generation are depicted in Figure 7.

```

db.execSQL("insert into test values(null,'Search Armstrong number: 371=3*3*3+7*7*7+1*1*1 \n" +
    "#include <stdio.h>; \n" +
    "#include <math.h>; \n" +
    "void main()\n" +
    " { int i,a,b,c; \n" +
    "\n" +
    "for(i=100;i=999;i++) \n" +
    " { a=i/100; \n" +
    "   _____1_____ // b=i%100/10; \n" +
    "   c=i%10; \n" +
    "if ( _____2_____ ) // a*a*a+b*b*b+c*c*c == i \n" +
    "   printf("%d is a Armstrong number!\n",i);\n" +
    " }\n" +
    " } ', '1', '1', null, null, null, 4)");

```

Figure 5. Insert of questions.

```

spinner = (Spinner) findViewById(R.id.spinner1); //controls for choice questions
up=(Button)findViewById(R.id.up); //button for previous question
down=(Button)findViewById(R.id.down); //button for next question
put=(Button)findViewById(R.id.put); //button for submission
mydb my=new mydb(this,"MyDb",null,1); //create database
//open a database that will be used for reading and writing
db=my.getWritableDatabase();
//query for choice questions
cursor2=db.rawQuery("select * from test where cid=1", null);
//query for fill-in-the-blank questions
cursor3=db.rawQuery("select * from test where cid=2", null);
//query for true-or-false questions
cursor4=db.rawQuery("select * from test where cid=3", null);
//query for fill-in-the-program questions
cursor5=db.rawQuery("select * from test where cid=4", null);

```

Figure 6. Data query in Android-based C++ quiz system.

```

size=cursor2.getCount(); //total number of queries for choice questions
for(int i=0;i<10;i++) {
    // for each loop, a random number ranged [1,size+1) is generated
    // and assigned to r[i]
    r[i]=random.nextInt(size)+1;
    for(int j=0;j<i;j++) {
        if (r[j] == r[i]) {
            i--;
        }
    }
}
size=cursor3.getCount(); //total number of queries for fill-in-the-blank questions
for(int i=10;i<15;i++) {
    // for each loop, a random number ranged is generated and assigned to r[i]
    r[i]=random.nextInt(size)+1+cursor2.getCount();
    for(int j=10;j<i;j++) {
        if (r[j] == r[i]) {
            i--;
        }
    }
}
size=cursor4.getCount(); //total number of queries for true-or-false questions
for(int i=15;i<19;i++) {
    // for each loop, a random number ranged is generated and assigned to r[i]
    r[i]=random.nextInt(size)+1+cursor2.getCount()+cursor3.getCount();
    for(int j=15;j<i;j++) {
        if (r[j] == r[i]) {
            i--;
        }
    }
}
size=cursor5.getCount(); //total number of queries for fill-in-the-program questions
r[num-1]=random.nextInt(size)+1+cursor2.getCount()+cursor3.getCount()+cursor4.getCount();

```

Figure 7. Testing paper generation.

```

//timer display for the test
final Handler handler = handleMessage(msg) -> {
switch (msg.what) {
case 1:
second++;
if(second>60){
second=0;
minuter++;
if(minuter==5){
MainActivity.this.show();
}
}
if(minuter>60){
minuter=0;
hour++;}
if(hour>60){
second=0;
minuter=0;
hour=0;
}
String sec = "0" + second;
sec = sec.substring(sec.length()-2, sec.length());
String minu = "0" + minuter;
minu = minu.substring(minu.length()-2, minu.length());
String hou = "0" + hour;
hou = hou.substring(hou.length()-2, hou.length());
tv.setText(hou+":"+minu+":"+sec);
break;
}
super.handleMessage(msg);
};
task = new TimerTask(){
public void run() {
Message message = new Message();
message.what = 1;
handler.sendMessage(message);
}
};
Timer timer = new Timer(true);
timer.schedule(task,2000,1000);

```

**Figure 8.** Timer generation during the test.

#### F. Timer

The Android-based C++ quiz system is used to check students' mastery degree of C++ knowledge points and to improve the students' programming ability. In the general examination, there is always time limit on the test, so it is necessary to incorporate a timer in the Android-based C++ quiz system. To this end, the method *sendMessage(Message)* of class *Handler* is used as message handing and sending. Moreover, the method *run()* is called to achieve timing during the thread execution on the question-answering interface. The core codes of timer generation are listed in Figure 8.

#### G. Scoring

Scoring is one of the main functions for the Android-based C++ quiz system. When the answers are submitted, testing scores must be delivered to the examinee. To this end, the following steps are taken to score the test. Firstly, an array, which is used to store the questions in the generation of the paper, must be obtained. Secondly, the method *rawQuery()* is used to query the questions in the combined testing paper. Thirdly, the method *equal()* is used to check whether the answer matches the keys to the questions or not. If the match is successful, corresponding scores are gained according to the preset scoring rules. The core codes of scoring are depicted in Figure 9.

#### H. Incorrect Answer Display

If someone hopes to make progress,he/she must always learn from the mistakes made byhim/her. In the Android-based C++ quiz system, incorrect answer display is incorporated. If someone does not earn a good scores,he/she can back to the question-answering page and find where the mistakes happen. According to the mistakes, the examinee can refer to helper or materials and finally correct the mistakes. To this end, two methods *intent* and *finish* are always used in Android Studio. The difference between them is that the first opens a new page, while the second finishes the current page when backing. In the system development, the method *finish* is used to terminate the scoring page, namely show page. The core codes back to the question-answering page are depicted in Figure 10.

```

public void show(){
    for(int i=0;i<num;i++) {
        cursor3 = db.rawQuery("select * from test where _id=" + r[i], null);
        cursor3.moveToNext();
        if (i < 10) {
            if (cursor3.getString(6).equals(a[i])) {
                grade=grade+4;
                n++;
            }
        } else if(i<15){
            if (cursor3.getString(6).equals(s[i])) {
                grade=grade+5;
                n++;
            }
        } else if(i<19){
            if (cursor3.getString(6).equals(s[i])) {
                grade=grade+5;
                n++;
            }
        } else if(i==num-1){
            if(cursor3.getString(2).equals(et1[i])){
                grade=grade+5;
                n++;
            }
            if(cursor3.getString(3).equals(et2[i])){
                grade=grade+5;
                n++;
            }
        }
    }
}
}
}

```

**Figure 9.** Scoring the test.

```

turn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        Show.this.finish();
    }
});

```

**Figure 10.** Back to the question-answering page.

#### IV. SYSTEM TESTING

The main purpose of the system testing is to verify whether or not the running results are good, whether or not the functions are perfect, whether or not the graphical user interfaces are friendly, and whether or not the performance is stable. To verify the Android-based C++ quiz system, testing is conducted on an Android virtual device emulator and a physical device, respectively.

##### A. Testing Platform

The hardware of the testing platform mainly consists of a desktop and Android-based smartphone. The desktop is configured with Intel(R) Core(TM) i5-4590 CPU, running at frequency 3.3 GHz, and 4.0 GB memory. The smartphone is configured with Kirin 659 CPU, 4.0 GB running memory, and 64 GB storage.

The software of the testing platform mainly refers to the operating system and application software on both the desktop and the smartphone. The software installed on the desktop are mainly 32-bit Windows 7 operating system and Android Studio 2.0. The software installed on the smartphone are Android-based operating system and exam app, developed by us. The detailed configuration of hardware and software are listed in Table I and Table II, respectively.

##### B. Testing Results

There are different types of software testing, but the main two types are white-box testing and black-box testing. The former is used to know exactly what is going on inside the program, while the latter is used to examine the functionality of an application without peering into its internal structures or workings. In the process of testing, black-box tests in AVD emulator and physical device are performed respectively to test all the main functions of the Android-based C++ quiz system. Testing of choice questions is shown in Figure 11, testing of fill-in-the-blank questions in Figure 12, testing of true-or-false questions in Figure 13, testing of fill-in-the-program in Figure 14 and testing of scoring in Figure 15, respectively.



TABLE I: Hardware configuration of the testing platform.

Device	Model	CPU	Memory	Storage
Desktop	Lenovo R4900D	Intel(R) Core(TM) i5-4590	4.0GB	500GB
Smartphone	Honor 7x	Kirin 659	4.0GB	64GB

TABLE II: Software configuration of the testing platform.

Device	Operation System	Application
Desktop	32-bit Windows 7	Android studio 2.0
Smartphone	Android 8.0.0	Exam app

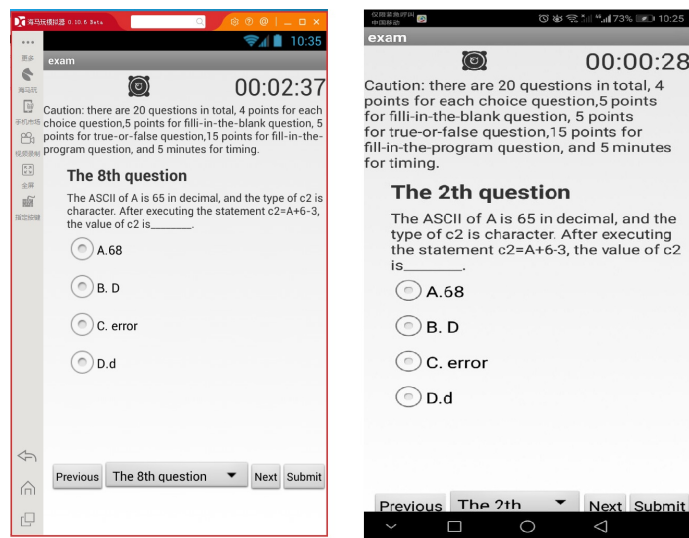


Figure 11. Testing of choice question in emulator (L) and physical device (R).

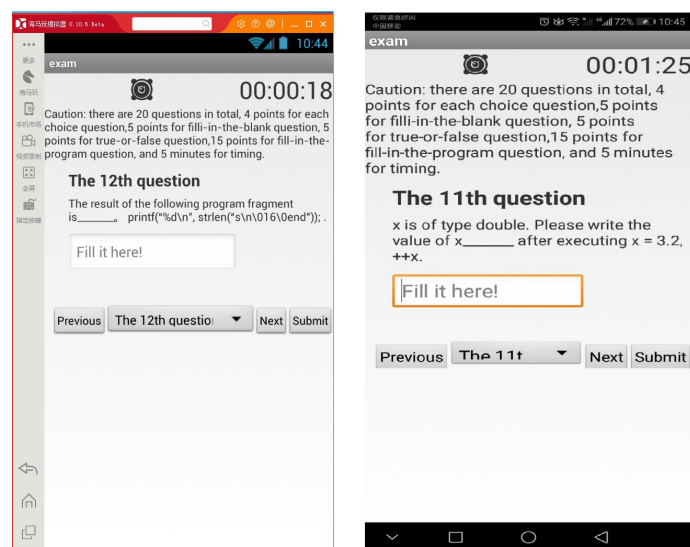


Figure 12. Testing of fill-in-the-blank question in emulator (L) and physical device (R).

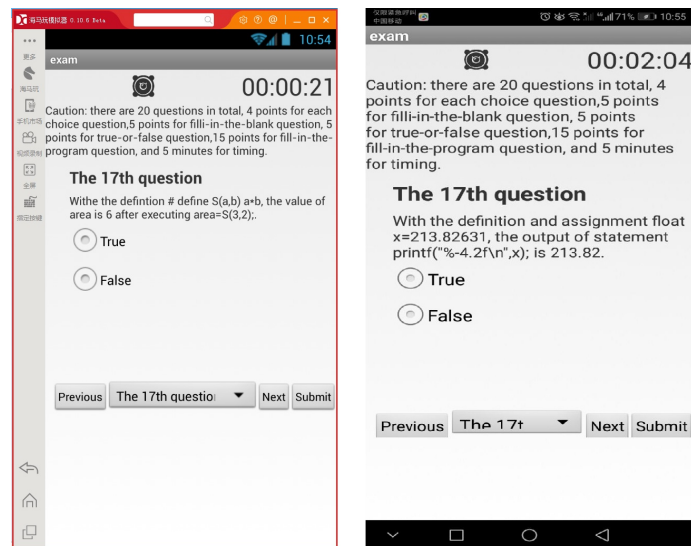


Figure 13. Testing of true-or-false question in emulator (L) and physical device (R).

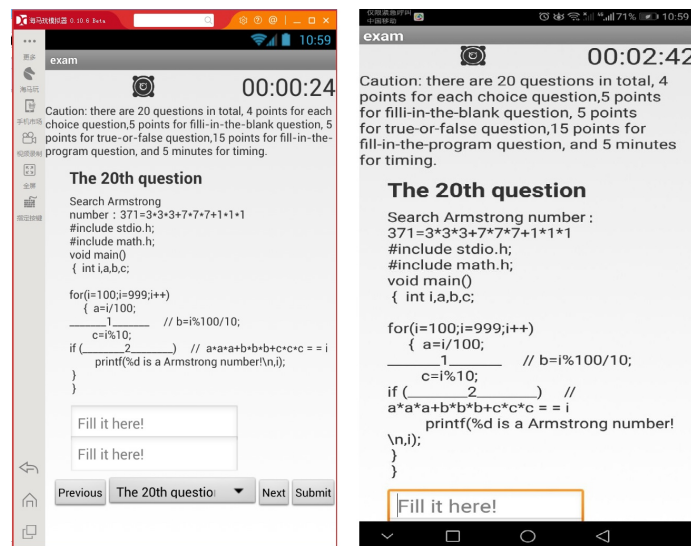


Figure 14. Testing of fill-in-the-program question in emulator (L) and physical device (R).

## V. CONCLUSIONS

To improve the college students' programming ability and strengthen China's competitiveness of software service outsourcing in the world, a portable Android-based C++ quiz system is designed and implemented. The Android-based C++ quiz system can randomly select four types of questions from the question bank to generate a testing paper. Questions including choices, fill in the blanks, true or false, and fill in the program can be added to the question bank for being randomly chosen to generate a distinct paper for each round testing. To well evaluate the quiz results, a timer and scoring function are included in the system. Testing results on the virtual emulator and real physical Android-based smartphone show that the Android-based C++ quiz system runs well and can be used as an assistant learning tool in C++ programming teaching.

## ACKNOWLEDGMENT

The authors would like to thank the valuable comments and suggestions of the anonymous reviewers. This work is supported by the 2015 Key Research Subject of Hubei Provincial Education Science Planning under Grant No. 2015GA038 and Teaching Research Project of Hubei Engineering College under Grant No. 2015A37, and the 2015 Research Project of Provincial Teaching Reform in Hubei Universities under Grant No. 2015400.

## REFERENCES

- [1] TIOBE TIOBE index for C++ (2017).[Online]. Available: <https://www.tiobe.com/tiobe-index>, Accessed: December 18, 2018.

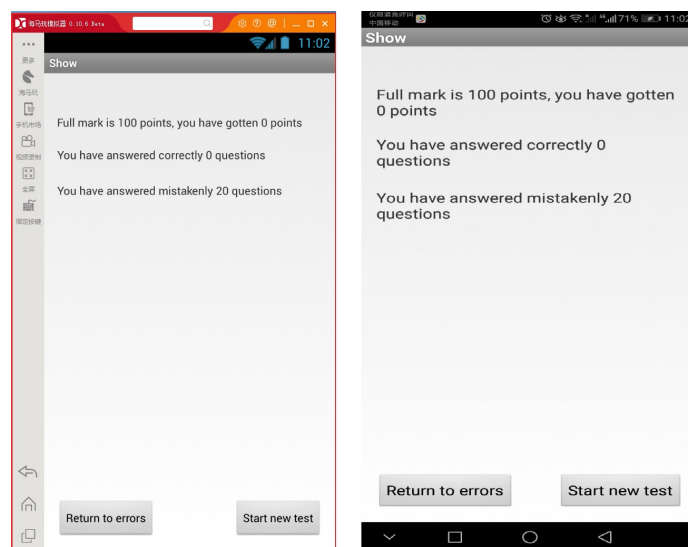


Figure 15. Testing of scoring in emulator (L) and physical device (R).

- [2] Donald Hsu, "Global Outsourcing to Engineer Retraining: New Skills for New Jobs in a Modern World," in proc. of *e-leader*, Bangkok, January 2-4, 2008.
- [3] Top 10 best applications written in C/C++.[Online].Available: <http://www.mycplus.com/featured-articles/top-10-applications-written-in-c-cplusplus/>,Accessed: 18 Dec. 2018.
- [4] Liang Tyng-Yeu, Li Hung-Fu, and Chen Yu-Chih, "A ubiquitous integrated development environment for C programming on mobile devices," In proc. of *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (DASC2014)*, PP.184-189, Dalian, China, August 2014.
- [5] Suhair Amer, "Approving an online C++ introductory programming course via Quality Matters through informal reviewers," in Proc. of *International Conference on Computational Science and Computational Intelligence*, pp.344-349, 2016.
- [6] U. Kokaew, S. Arch-Int, P. Suesawaluk, "Design and Development of an Integrated Online System Support for C/C++ Programming," *Thammasat International Journal of Science and Technology*, pp.47-58, vol. 20, no.1, 2015.
- [7] Tian Runhu, "The application of a micro-blog platform in the teaching of C programming," *World Transactions on Engineering and Technology Education*, vol. 13, no. 4, pp. 539-543, 2015.
- [8] Liang Tyng-Yeu, Li Hung-Fu, Chen Yu-Chih and Lin Yu-Jie, "A mobile integrated development environment for C programming," In Proc. of *International Journal of Embedded Systems*, vol. 8, pp. 493-503, 2016.
- [9] Jamie Peck, *Offshore Exploring the Worlds of Global Outsourcing*, Oxford University Press, 2017.
- [10] Wenli Geng and Yingsong Hu, "SWOT Analysis of China Software Service Outsourcing," in Proc.of *International Conference on Software Engineering & Service Science (ICSESS)*, pp.691-694, Beijing, China, July 2011.
- [11] China Computer Software Industry Report 2014-2017.[Online]. Available: <http://www.docin.com/p-798690021.html>,Accessed: December 18, 2018.
- [12] DIGITIMES,[Online]. Available: <http://www.digitimes.com/news/a20180131VL200.html>,Accessed: 18 Dec. 2018.
- [13] Java SE 7 Features and Enhancements.[Online]. Available: <https://www.oracle.com/technetwork/java/javase/jdk7-relnotes-418459.html>,Accessed: December 18, 2018.
- [14] Java Development Kit.[Online]. Available: [https://en.wikipedia.org/wiki/Java\\_Development\\_Kit#cite\\_ref-2](https://en.wikipedia.org/wiki/Java_Development_Kit#cite_ref-2),Accessed: December 18, 2018.
- [15] Neil Smyth, *Android Studio 3.0 Development Essentials (8 Edition)*, Createspace Independent, 2017.