“ON LINE” TESTING FOR SOFTWARE DESIGN

Marian Gheorghe, Mihaiu Ilie Mircea
Dumitrascu Eugen, Enescu Nicolae-Iulian

University of Craiova
Faculty of Automatics, Computers and Electronics
str. Lapus no.5, 1100 Craiova-Romania
gheorghe.marian@comp-craiova.ro, mihaiu@nt.comp-craiova.ro,
eugen.dumitrascu@comp-craiova.ro, nicu.enescu@comp-craiova.ro

Abstract. The purpose of this paper is to present the new “on line’ interactive testing methods for software developers. First a new thematic of software module is proposed and then a step-by-step solution must be implemented. Each step in design procedure has some proposed solution and the person who follows the test must choose the correct answer. The second variant of testing procedure is to choose a correct sentence of the software module using a list of sentences presented above. There are many possible variants and a complete response must present all the possible variants. The proposition for thematic software module isn’t optimized for the simplest solution or for most efficient solution. Our proposition is only for testing the correct sequence of a software program and the correct using of some instructions and commands.

Keywords: “on line” testing, software design testing, interactive testing, e-Learning, e-Testing.

1. INTRODUCTION

There is an increasing evidence that the ICT skills are a very important role in each economic and scientifically activities. Professional ICT skills are imperious claim to the marketplace, and the new testing methods for ICT skills are developed. The main elements driving the demand of such skills are the growing importance of the telecommunication technology and the Internet technology. The “on line” testing is an advanced concept that helps companies to find the talent workers. There are many international websites where anybody can do a test or obtained a skills certificate [Russielo 2000, Kohut 2003, OECD 2001]. There are different ICT skills [OECD 2003]:

-Professional ICT skills: ability to use advanced ICT tools, and/or to develop, repair and create such tools,

-Applied ICT skills: ability to apply ICT tools in general workplace settings (in non –ICT jobs),

- Basic ICT skills or “ICT literacy”: ability to use ICT basic tasks as a tool for learning.

Some college and universities in US are now providing certification-related training [Nagel 2003], either al a Carte or as part of a larger degree program. Also smart training providers are now working in partnership with higher education.

In this new marketplace the need for qualified workers is very high and there are required appropriate methods and techniques to check the skills and competences of individuals. Whenever a person feels that it is ready for testing it can apply for the test on the Internet.

A typical Internet questionnaire contains 30-40 questions with up to 2 minutes assigned for giving the answer for a question. We think that this questionnaire can’t verify all skills for the testing person. Techniques and procedures for testing must be diversified. In this paper we propose new techniques for testing persons in the field of software design.

Each software program can be considered like an assembled piece composed by many small pieces. Small pieces are instructions, directives, some symbols and these pieces must be assembled in a specified order to obtain a workable program. The software programmer must know all pieces and the steps to follow for assembling these pieces. For testing software design we propose two variants of tests:

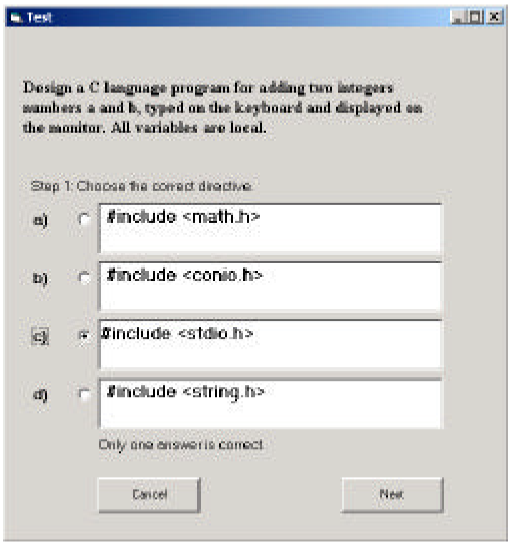
- The step by step testing with multiple choose for each step
- The correct sentence choosing for the entire software packet.

For the beginners the step-by-step test is recommended and for testing workers with some experience the correct sentence test is recommended.

2. STEP-BY-STEP TESTING METHOD

This step-by-step testing method is an interactive software design by follows a step-by-step procedure. First the main statement is proposed then the individual who follow the test must choose the correct sentence of instructions or directives for each step. If the correct sentence is choosing, then the next step follows. If the wrong sentence is choosing a message alert is displayed and another choose must be done from the list of possible answers. If the tested person chooses always the correct answer he/she finish the test quickly without any alert messages. All wrong answers are countered and at the end of the test are displayed. The final mark is obtained from a correspondence table between marks and wrong answers.

For better understand this procedures let’s give an example.



Design a C language program for adding two integers numbers a and b, typed on the keyboard and displayed on the monitor. All variables are local.

Step 1: Choose the correct directive

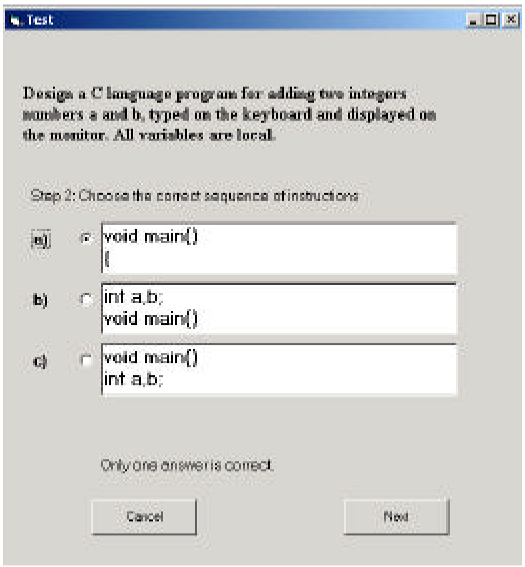
a) ○ #include <math.h>

b) ○ #include <conio.h>

c) ● #include <stdio.h>

d) ○ #include <string.h>

Only one answer is correct



Design a C language program for adding two integers numbers a and b, typed on the keyboard and displayed on the monitor. All variables are local.

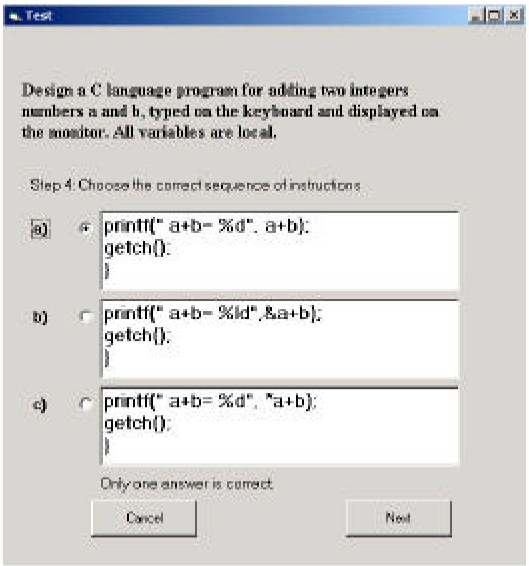
Step 2: Choose the correct sequence of instructions

a) ●
```
void main()
{
```

b) ○
```
int a,b;
void main()
```

c) ○
```
void main()
int a,b;
```

Only one answer is correct



Design a C language program for adding two integers numbers a and b, typed on the keyboard and displayed on the monitor. All variables are local.

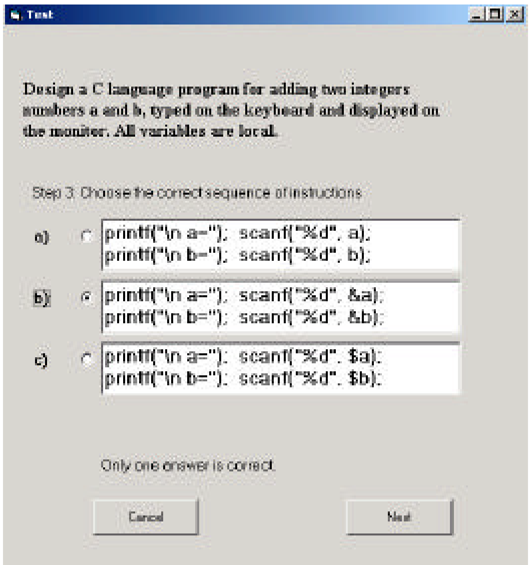
Step 4: Choose the correct sequence of instructions

a) ●
```
printf(" a+b= %d", a+b);
getch();
}
```

b) ○
```
printf(" a+b= %ld",&a+b);
getch();
}
```

c) ○
```
printf(" a+b= %d", *a+b);
getch();
}
```

Only one answer is correct.



Design a C language program for adding two integers numbers a and b, typed on the keyboard and displayed on the monitor. All variables are local.

Step 3: Choose the correct sequence of instructions

a) ○
```
printf("\n a="); scanf("%d", a);
printf("\n b="); scanf("%d", b);
```

b) ●
```
printf("\n a="); scanf("%d", &a);
printf("\n b="); scanf("%d", &b);
```

c) ○
```
printf("\n a="); scanf("%d", $a);
printf("\n b="); scanf("%d", $b);
```

Only one answer is correct.

The final mark can be obtained from a correspondence table like the next.

Correspondence table:

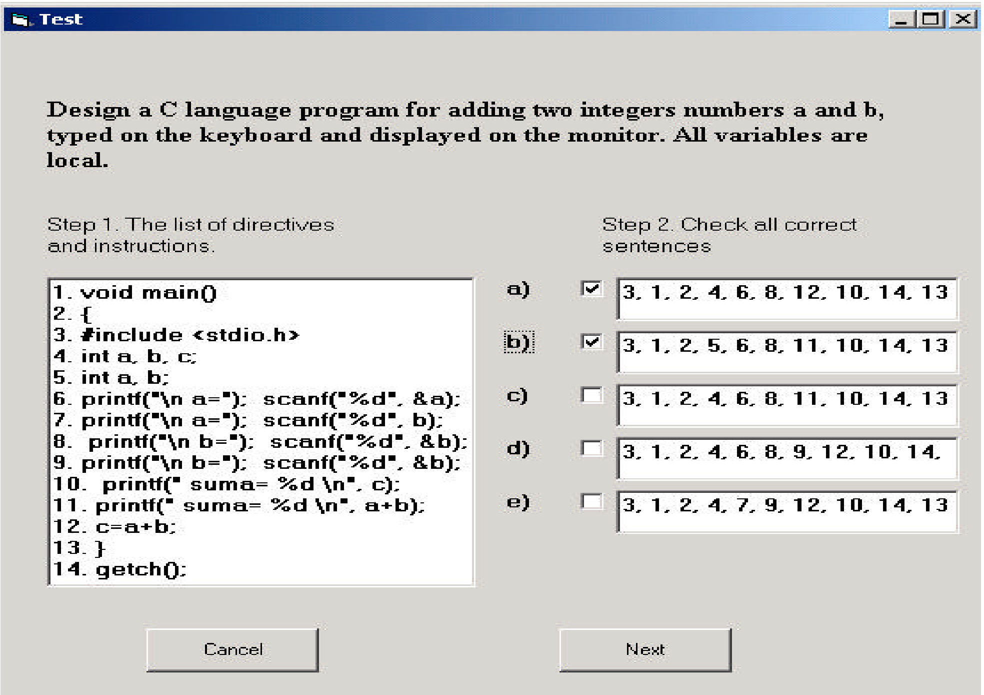
Wrong Answers	Mark	Wrong Answers	Mark
0	10	5	5
1	9	6	4
2	8	7	3
3	7	8	2
4	6	9	1

The tested person can know or can't know correspondence table.

Some tests can have two or more correct answers and the final program can have many variants. For each problem the test designer can choose wrong answers very closely to the correct answer to obtain a difficult test. This category of test is very useful for beginners in software design.

3. THE CORRECT SENTENCE TESTING METHOD

The second testing method verifies if the tested person can find all the variants of software composed by many directives and instructions. First, it is presented a list of directives and instructions and then the possible variants of grouping these instructions and directives for solve the problem. The final mark is a function of the correct variants selected by the tested person. We present an example of such test.



Design a C language program for adding two integers numbers a and b, typed on the keyboard and displayed on the monitor. All variables are local.

Step 1. The list of directives and instructions.

```
1. void main()
2. {
3. #include <stdio.h>
4. int a, b, c;
5. int a, b;
6. printf("\n a="); scanf("%d", &a);
7. printf("\n a="); scanf("%d", b);
8.  printf("\n b="); scanf("%d", &b);
9. printf("\n b="); scanf("%d", &b);
10.  printf(" suma= %d \n", c);
11. printf(" suma= %d \n", a+b);
12. c=a+b;
13. }
14. getch();
```

Step 2. Check all correct sentences

a) ☑ 3, 1, 2, 4, 6, 8, 12, 10, 14, 13

b) ☑ 3, 1, 2, 5, 6, 8, 11, 10, 14, 13

c) ☐ 3, 1, 2, 4, 6, 8, 11, 10, 14, 13

d) ☐ 3, 1, 2, 4, 6, 8, 9, 12, 10, 14,

e) ☐ 3, 1, 2, 4, 7, 9, 12, 10, 14, 13

The final mark is a function of the number of correct sentences founded.

4. CONCLUSIONS

The testing methods for software design must be diversified for checking the real skills of a tested person. We present in this paper two methods for verifying programming skills "on line". These testing methods are implemented to the testing site http://ctconline.ucv.ro

REFERENCES

1. Kevin Kokut," Essential Tech Skills for Every IT Pro", certcities.com, www.certcities.com, 2002

2. C.Luncan ,"AcademiaOnline.ro – primul Siustem de e-learning public roamanesc ",PC Magazine, Romania, Aprilie, 2003

3. M.I.Mihaiu et all, " The development of a testing and certification "on line" site", Annals of the University of Craiova,Electrical Engineering Serie, Yesr 26, no.26, 2002, Second part.

4. B. Nagel, "ANALYSIS: Reporst from Strategies 2003, State of the Industry&Trends for the Future", CertCities.com, June ,26, 2003, http://certcities.com

5. OECD, V. Lopez-Bassols, G.Vickery,"ICT Skills and Employment-Workink Party on the Information Economy", Paris,October,11-12, 2001, www.e-skills-summit.org

6. M.Russielo, "The Rise of online Certification", January, 2000, www.brainbench.com

7. M.Waite, et all ,"C Primer Plus", Howard W. Sans By Ca, 1986

8. Jamsa K., Klander L.- Totul despre C si C++. Teora 2002

9. Schildt H. – C manual complet. Teora 1998

10. Schildt H. – C++ manual complet. Teora 1998