A test system for checking and evaluation the students' programming knowledge

Nikolay Kirov

Computer Science Department, New Bulgarian University, Sofia, Bulgaria

nkirov@nbu.bg

Introduction

- The article describes the preparation and implementation of multiple choice type tests for checking the students' knowledge of programming.

- Justification of the chosen approach, methodology and technical details are discussed.

- Software for generating tests and analysing of results is proposed.

Contents

- Terminology

- Special multiple choice test

- Disadvantages

- Test generator

- Test checker

- Example

- Conclusion

Terminology

- The test consists of items.

- The stem is the introductory question or statement at the beginning of each item.

- The stem is followed by the options.

- The options consist of

  - answers − the correct options, and
  - distractors − the incorrect options.

- The items are stored in an item bank.

- Individual test consists of fixed number of items with fixed number of options.

Special multiple choice test

Our issue is a specific multiple choice test in several directions:

- The number of options is fixed to 4 for all items.

- The number of correct options (answers) of an item may be any number in the interval [0,4], the remaining options serve as distractors (multiple response items);

- The student is directed to identify each answer and each distractor. He/she has 3 choices of response:

  - yes, i.e. I know it is an answer;
  - no, i.e. I know it is a distractor;
  - nothing, i.e. I do not know whether the option is an answer or a distractor.

- Correct response (yes or no) wins a point and incorrect response loses one point (penalty point) in the total score.

Arguments and advantages

- Some arguments for these nonstandard characteristics (0, 1 or many answers per item, 3-valued logic and penalties) of our multiple choice test for students in programming can be found in
  N. Kirov, A System for Assessing the Knowledge and Skills of Students in Computer Programming, CSECS'2013.

- About penalties, for example, the SAT test system removes a quarter point from the test taker's score for an incorrect answer.

- The advantages of multiple choice tests are well known and will not be discussed here. More important is how to avoid the disadvantages of this type of testing.

The most serious disadvantage is the limited types of knowledge that can be assessed by multiple choice tests. Multiple choice tests are best adapted for testing well-defined or lower-order skills.

Introductory courses on programming possess the characteristics:

- the ideas, methods and rules in programming are well-defined;
- an essential part of programming skills is the low-order knowledge: syntax, simple constructions, etc.

Modifications to the standard multiple choice test contribute to setting more sophisticated and complex questions that check students' knowledge at a higher level.

This test is not alone and it is not sufficient for a complete assessment of student programming skills.

Another disadvantage of multiple choice tests is possible ambiguity in the student's interpretation of the item.

We use the following steps in avoiding this disadvantage:

- before the test time

  - At least one week before the date of the test all original stems and two example options per item (an answer and a distracton) are published online on the course website. Thus, the students have the opportunity to learn about the test in advance.
  - A few days before the test, a general advice is organized on which the issues of the test are discussed.

Another disadvantage of multiple choice tests is possible ambiguity in the student's interpretation of the item.

- during the completion of the test

  - During the test time students can use lectures, textbooks and any other printed materials.
  - Sometimes students are allowed to use a computer, compiler, and the Internet.
  - If a student have a questions about ambiguities in the test, the instructor can answer the questions personally.

- after the verification of the tests

  - After checking the tests, they are returned to students.
  - Each student should carefully check the test in order to determine whether he/she agrees with the errors noted.
  - If something is not clear she/he discuss the case with the instructor.
  - Our practice is to accept the students' opinions who have different interpretations and possibly to increase the test points.

Another disadvantage of multiple choice examinations is that a student who is incapable of answering a particular question can simply select a random answer and still have a chance of receiving a mark for it.

- We estimate a test of `M` items with `x` points as follows:
```
if ( x/M/4*100 >= 50 ) e = 3;
if ( x/M/4*100 >= 60 ) e = 4;
if ( x/M/4*100 >= 76 ) e = 5;
if ( x/M/4*100 >= 90 ) e = 6;
else e = 2;
```

- To calculate the probability of passing the test using random method, we choose a test with 10 items and maximum 40 points..

- Probability is 0.11% in case the student has noted all the options yes or no (given the answer to all questions).

- Probability is 0.0034% in case the student marked randomly options with yes, no or nothing.

Test generator

- Preparation of the test begins with selecting items – stems and options and put them into item bank.

- At least 10 items each having at least 5-6 possible answers should be completed and stored as a text file in a particular format. This file is the input to `test_generator`.

- `test_generator` generates individual tests using random distribution of both items and their options.

- Each individual test consists of 10-20 items with 4 options, marked as a), b), c), d).

- The output plain text file (`out.tex`) (in LaTeX format) contains all individual tests.

- The second output file (`tab.tex`) is a table for checking the tests.

- The third file (`data.txt`) is a copy of the input file with additional data for the generated individual tests.

Test generator

Test generator Example: `out.tex` after compilation (`.pdf`):

NETB151 Object-Oriented Programming (C++)

Test No. 2361

---

1. We have the following classes:

```
class Point {
... };
class Circle : public Point {
... };
```

Mark the correct/incorrect assertions about the classes Point and Circle.

a) The member functions of the class Circle have access to any data member of the class Point.

b) The data members of the class Point are present in each object of the class Circle.

c) The member functions of the class Point have access to any data member of the class Circle.

d) The class Point is the base class and the class Circle is the derived class.

---

2. Mark the correct/incorrect assertions about pointers.

a) The value of an array variable is a pointer to the starting element of the array.

b) Finding the value to which a pointer points is called dereferencing.

c) The * operator locates the value to which a pointer points.

d) The value of a pointer must be an address in the heap memory.

```
b) double py = new double;
c) int* pn = new int;
d) Time* pt1 = new Time(10,0,0);
```

---

7. Mark the correct/incorrect assertions about inheritance hierarchy of stream classes.

a) The fstream class derives from ofstream.

b) The ifstream class derives from istream.

c) The fstream class derives from ifstream.

d) The iostream class derives from ifstream and ofstream.

---

8. Mark the correct/incorrect assertions about stream classes, objects and member functions.

a) The << operator is defined for ostream objects.

b) The object cin belongs to a class that is derived from ifstream.

c) The object cin belongs to a class that is derived from istream.

d) The close member-function is defined for fstream objects.

---

9. Suppose the class D inherits from B. Let b be an object of the class B, d be an object of the class D, pb be a pointer of the class B and pd be a pointer of the class D. Which of the following assignments are legal?

```
a) pd = pb;
```

Test generator

Example: `tab.tex` after compilation (`.pdf`):

Test 1 – NETB151, 27.3.2014

2319	1.cd	2.c	3.acd	4.ad	5.abcd	6.abcd	7.	8.bcd	9.bcd	10.bcd	11.bd	(27)
2326	1.bcd	2.bc	3.ab	4.cd	5.b	6.bcd	7.acd	8.d	9.c	10.acd	11.acd	(24)
2333	1.acd	2.ad	3.bcd	4.ad	5.bd	6.a	7.ac	8.	9.abc	10.bc	11.ad	(22)
2340	1.acd	2.abd	3.ad	4.abd	5.abcd	6.acd	7.abcd	8.d	9.	10.d	11.c	(25)
2347	1.	2.ac	3.abc	4.cd	5.bcd	6.bd	7.acd	8.abc	9.ad	10.d	11.ac	(23)
2354	1.	2.bcd	3.c	4.	5.bd	6.b	7.a	8.cd	9.abc	10.bcd	11.bcd	(19)
2361	1.bd	2.abc	3.a	4.bcd	5.abc	6.cd	7.b	8.acd	9.cd	10.cd	11.acd	(25)
2368	1.bd	2.bd	3.	4.abd	5.ad	6.ab	7.b	8.acd	9.acd	10.	11.c	(19)
2375	1.ad	2.bd	3.bcd	4.acd	5.a	6.abc	7.d	8.acd	9.d	10.abc	11.c	(23)

Test checker

Checking test can be carried out

- manually − using the table (`tab.tex`) generated by `test_generator` or

- automatically − by `test_checker`.

The input of the students' answers can be done:

- manually − using the user interface of `test_checker` or

- automatically − using a special template for students' answers and scanner.

The program creates a text file (`save.txt`), containing audited tests.

Test checker

```
4
q11
a no +
B yes +
c no +
D yes +
4
Total 24 ( 6 pt )
Test No. 2326
q1
a yes -
B yes +
C yes +
D yes +
2
q2
a no +
B no -
C no -
d no +
0
q3
A yes +
```

Example: `save.txt`:

Test checker

After entering all the tests, `test_checker` gives the results − for each option of each item of the test bank calculates two sets of numbers. The set $A = \{a, a_1, a_2, a_3\}$ represents all the tests and the set $B = \{b, b_1, b_2, b_3\}$ represents the tests which collect at least a half of maximum points, i.e. .

- a, b − the number of individual tests which contain the corresponding item and four of its options;

- a_1, b_1 − the number of tests with the correct response;

- a_2, b_2 − the number of tests with the incorect (opposite) response;

- a_3. b_3 − the number of tests without response.

$a_1 + a_2 + a_3 = a$, $b_1 + b_2 + b_3 = b$, $b \le a$ and $b_i \le a_i$ for $i = 1, 2, 3$.

Test checker

Test checker, v.1.1, 5.02.2014

Test 3749   1

a
A)   ○ 0   ○ yes   ⦿ no   -

b
B)   ○ 0   ⦿ yes   ○ no   +

c
c)   ○ 0   ⦿ yes   ○ no   -

d
d)   ○ 0   ⦿ yes   ○ no   -

Total   -2   Quest   -2
Exam 10   Points   0

Load   Save

Calculate

Example

First test of Object-oriented programming (second semester).

- Test bank consists of 12 item.

- The numbers of options are: (14,16,19,16,12,7,12,16,10,9,16,22).

- Any individul test consists of 11 items (maximum 44 points).

- We have 19 individual tests.

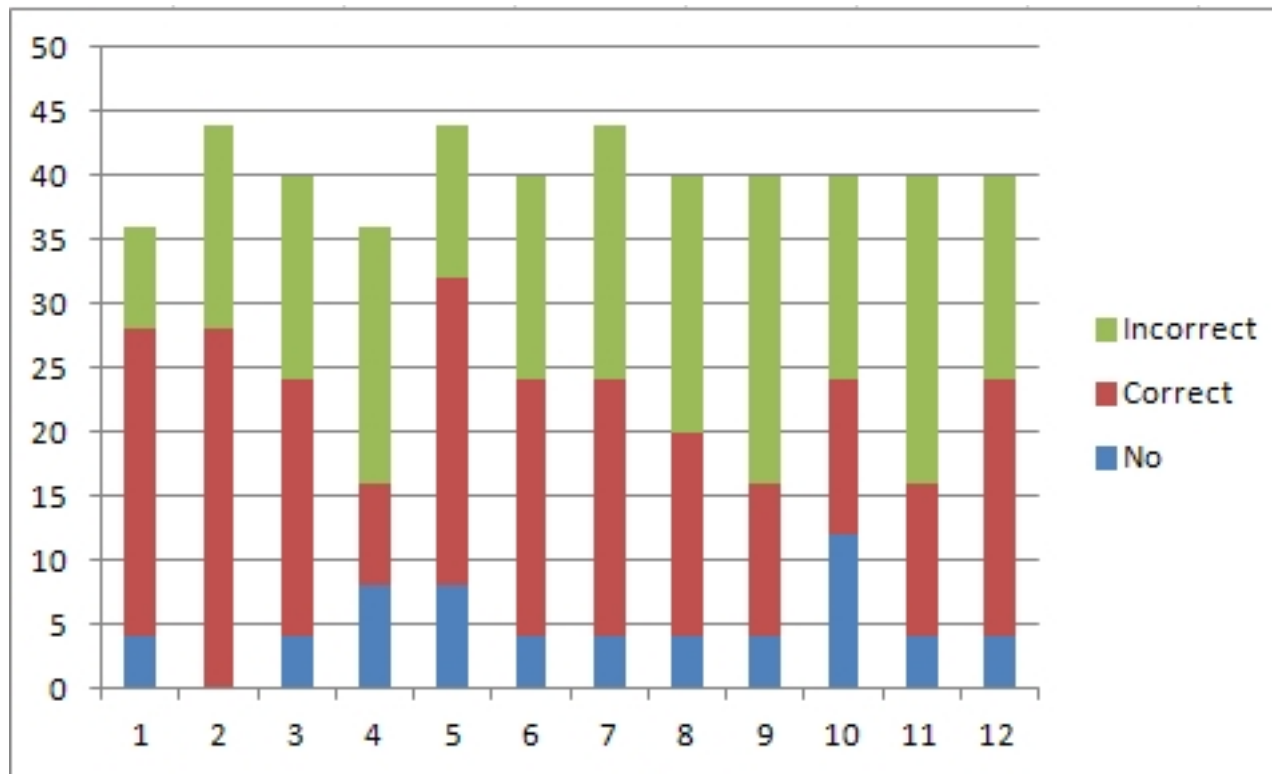## Example

Items for 19 individual tests – X items, Y results.

- Incorrect – the number of tests with the incorect (opposite) response.
- Correct – the number of tests with the correct response.
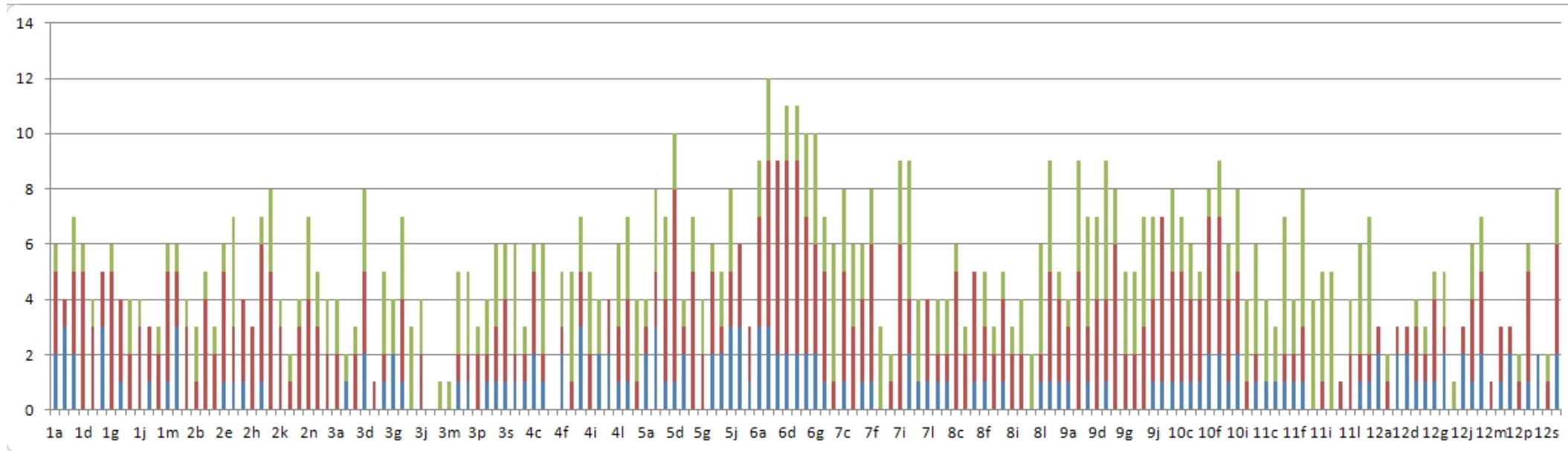- No – the number of tests without response.
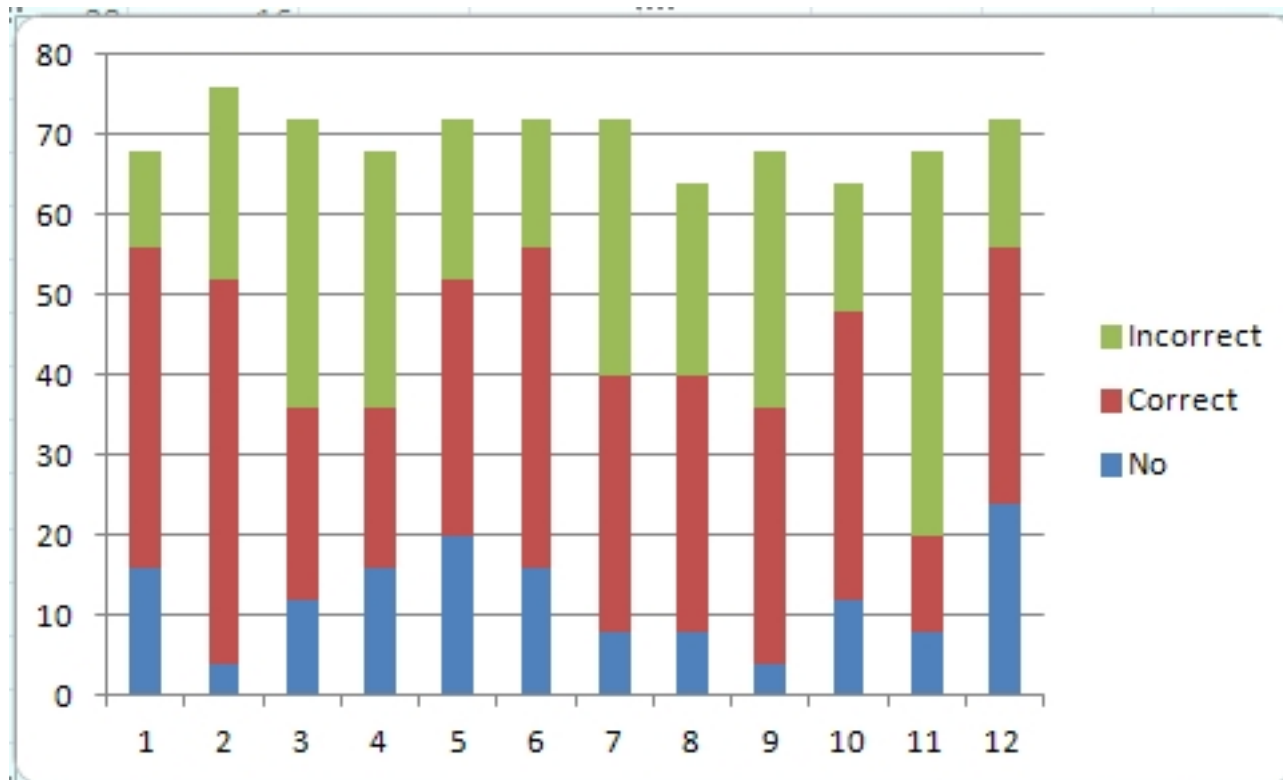
Example

Items for 11 individual tests, which are collected at least a half (22) of maximum points (44).

- Incorrect − the number of tests with the incorect (opposite) response.
- Correct − the number of tests with the correct response.
- No − the number of tests without response.

Example

Options for 19 tests − − X options, Y results.

Conclusion

The software is written in C++ using Qt − cross-platform application and UI development framework (`http://qt.digia.com/`).

- `https://github.com/nkirov/tests_generator`

- `https://github.com/nkirov/tests_checker`

github.com/nkirov/tests_generator   github.com/nkirov/tests_checker

Thank you for your attention.

nikolay.kirov.be   nkirov@nbu.bg