

National Exams December 2013

04-Soft-A6, Software Quality Assurance

3 hours duration

NOTES:

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit with the answer paper, a clear statement of any assumptions made.
2. This is an OPEN BOOK EXAM. Candidates may use any non-communicating calculator.
3. FIVE (5) questions constitute a complete exam paper. The first five questions as they appear in the answer book will be marked.
4. Each question is of equal value.
5. Most questions require short written answers. Clarity and organization of the answer are important, but full sentences are NOT required. Be sure to bullet lists and ideas wherever possible.

1.
 - a) What is software quality assurance? Identify and describe very briefly (1-2 lines) activities, which are common for any model or standard for software quality assurance.
 - b) List two software quality assurance (SQA) standards (or software process standards which involve SQA) you know and describe the purpose of them very briefly (1-2 lines).
 - c) Is SQA an umbrella activity in software development process or this is a phase within the process?

2.
 - a) Identify and describe briefly (3-4 lines) major SQA activities during the software development process.
 - b) What is agile software development?
 - c) Compare and contrast SQA activities in two cases: (1) if you use the Waterfall model and (2) agile development.

3.
 - a) List testing strategies you know for a relatively large software system. Compare and contrast them.
 - b) What is test driver and test stub? How test drivers and stubs are used in different test strategies?

- 4.)
 - a) Describe unit testing techniques you know. Compare and contrast them. List types of test approaches for each technique.
 - b) What are the four equivalence classes in the equivalence partitioning test?

5. Assume you test an access control (password verification) unit with two edit boxes: one for username, another one for password (like we have for the most secured systems). The username should be 4-to-8 characters length, and should be in the system database. The password should be 8-10 characters length, and is required to be with at least one capital letter, one number, and one special character. Create a black-box test for this unit.

6. Assume you test a dummy unit in Appendix A. Create a test for this unit using the basis path testing approach. Hint: it is important to focus on the functions' hierarchy.

7.
 - a) What is the difference between software verification and validation? Illustrate your answer by the use of the program in question #6.
 - b) What is system testing, alpha-testing, and beta-testing?

8. Describe any test automation tool you know or have used (10 lines). How this tool can be helpful for agile software development. If you are not familiar with any, explain how a version control system, for instance CVS, can be useful to improve software quality.

Appendix A

```

// Filename:
// A simple in-memory property database program. Through a menu,
// the user decides if he or she wants to see a property
// database on the screen or search for a specific property.
#include <iostream.h>
#include <string.h> // For strcmp()

void DisplayMenu();
int GetAnswer();
void DisplayProps(char * code[], float price[],
                 char * addr[], float commPer[]);
void SearchProps(char * code[], float price[],
                 char * addr[], float commPer[]);

// Eight properties maximum due to next constant
int const NUM = 8;

void main()
{
    int ans;
    // Define the program's data in parallel arrays
    // A code that uniquely identifies each property
    char * code[NUM] = { "231DV", "821WQ", "199OI", "294JU",
                        "901RE", "829BN", "483LQ", "778AS" };
    // The price of each property
    float price[NUM] = { 89432.34, 123029.34, 321293.95,
                        214293.20, 68402.92, 421034.53,
                        232456.54, 432123.40 };
    // The address of each property
    char * addr[NUM] = { "919 N. Elm", "2202 West Sycamore",
                        "7560 E. 26th Pl.", "213 W. 104th Ave",
                        "123 Willow Rd.", "5629 S. 188th",
                        "45 North Harvard", "17093 Lansford" };
    // The broker's commission on each property
    float commPer[NUM] = { .072, .07, .065, .091,
                          .078, .0564, .102, .0834 };
    do
    {
        DisplayMenu();
    }
}

```

```

ans = GetAnswer();

switch (ans)
{ case (1) : { DisplayProps(code, price, addr, commPer);
              break; }
  case (2) : { SearchProps(code, price, addr, commPer);
              break; }
  case (3) : { return;
              break; } // "unreachable code"
} // If user entered bad value, while loop will repeat
} while (ans != 3); // Keep looping until return takes over
return;
}
//*****
void DisplayMenu()
{ // Display a menu for the user
  cout << endl << endl;
  cout << "\t\t** Property Database Menu **" << endl << endl;
  cout << "Here are your choices:" << endl << endl;
  cout << "\t1. Look at the property listing" << endl;
  cout << "\t2. Search for a property by its code" << endl;
  cout << "\t3. Quit the program" << endl;
  cout << endl << "What is your choice? ";
  return;
}
//*****
int GetAnswer()
{ // Get the user's menu choice
  int ans; // Local variable also named ans
  cin >> ans; // Answer to menu
  cin.ignore(80, '\n');
  return (ans);
}
//*****
void DisplayProps(char * code[], float price[],
char * addr[], float commPer[])
{ // Display a list of properties
  int ctr; // for-loop control variable
  cout.precision(2);
  cout.setf(ios::showpoint);
  cout.setf(ios::fixed);
  for (ctr = 0; ctr < NUM; ctr++)
  {
    cout << endl << "Code: " << code[ctr]
      << "\t Price: $" << price[ctr] << endl;
    cout << "Address: " << addr[ctr] << endl;
  }
}

```

```

    cout << "Commission percentage: "
        << commPer[ctr] * 100.0 << "%" << endl << endl;
    if (ctr == 3) // Don't scroll off too fast
    {
        cout << "Press enter to continue...";
        cin.ignore(80, '\n');
    }
}
cout << "Press enter to continue...";
cin.ignore(80, '\n');

return;
}
//*****
void SearchProps(char * code[], float price[],
char * addr[], float commPer[])
{ // Ask the user for a property code and display match
    int ctr; // for-loop control variable
    int found = 0; // Initially not found
    char buf[6]; // Code plus null zero size
    // Get the search key
    cout << "I'll now search for a specific property." << endl;
    cout << "What is the property's code? ";
    cin.getline(buf, 6);
    for (ctr = 0; ctr < NUM; ctr++)
    {
        if (!strcmp(code[ctr], buf))
        {
            cout << endl << "Code: " << code[ctr]
                << "\t Price: $" << price[ctr] << endl;
            cout << "Address: " << addr[ctr] << endl;
            cout << "Commission percentage: "
                << commPer[ctr]*100.0
                << "%" << endl << endl; // Show as a percent
            found = 1;
            break;
        }
    }
    if (!found)
    {
        cout << endl << "* I'm sorry, but I don't find code "
            << buf;
    }
    return;
}

```

04-Soft-A6 Software Quality Assurance

Marking Scheme

1. a) 3 marks
b) 5 marks
c) 2 marks
 2. a) 5 marks
b) 2 marks
c) 3 marks
 3. a) 5 marks
b) 5 marks
 4. a) 5 marks
b) 5 marks
 5. 10 marks
 6. 10 marks
 7. a) 5 marks
b) 5 marks
 8. 10 marks
-