SFWR ENG 2S03/COMP SCI 2S03

Day Class
Duration of examination: 50 minutes
McMaster University Midterm Examination

October 2011

This examination paper includes 5 pages and 9 questions. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator.

SPECIAL INSTRUCTIONS: This paper must be returned with your answers. Use of McMaster standard (Casio-FX991) calculator only is allowed.

1. Given the function

   ```
   int sum(int N) {
       int s = 0;
       for (int i = 1; i <= N; i++) {
           s += i;
       }
       return s;
   }
   ```

   (a) (1 mark) What is the lifetime of the variable `s`?
   The variable `s` persists when the function `sum` is called until it returns.

   (b) (1 mark) What is the scope of the variable `i`?
   The scope of `i` is the `for` block.

   (c) (1 mark) What is the value returned by `sum(4)`?
   10

2. Indicate the value and type of each of the following expressions:
   a. (2 marks) 19 / 5, value 3 and type `int`.
   b. (2 marks) Assuming the value of `n` (int) is 3, `n == 5`, value `false` and type `bool`.
   c. (2 marks) 18 * 0.5, value 9.0 and type `double`.
   d. (2 marks) Assuming the value of `x` (double) is -1.7, `(x > 0)? (x + 0.5) : (x - 0.5)`, value -2.2 and type `double`. 
3. (1 mark) Assuming the declarations:
    bool x, y;
    int *p;

Which of the following is NOT an lvalue:
    a. x || y    b. y    c. p    d. *p

Answer: a

4. (5 marks) Draw the diagram showing the contents of memory after each line of the following code:

    v1 = 10; v2 = 25; p1 = &v1; p2 = &v2;
    *p1 += *p2;
    p2 = p1;

Three diagrams (show both addresses and arrows):

![Diagram showing memory contents after each line of code](image-url)
5. (4 marks) Implement the following function that counts the number of spaces in a string.

```cpp
int CountSpaces(string str) {
    int nSpaces = 0;
    for (int i = 0; i < str.length(); i++) {
        if (str[i] == ' ') {
            nSpaces += 1;
        }
    }
    return nSpaces;
}
```

6. (a) (3 marks) Define an enumeration type `polygonT` consisting of the elements: Triangle, Square, Pentagon, Hexagon, Octagon.

```cpp
enum polygonT {
    Triangle,
    Square,
    Pentagon,
    Hexagon,
    Octagon
};
```

(b) (2 marks) Change the definition so that internal representation for each constant name corresponds to the number of sides for that polygon. (Note, a pentagon has five sides, hexagon has six sides, and octagon has eight sides.)

```cpp
enum polygonT {
    Triangle = 3,
    Square,
    Pentagon,
    Hexagon,
    Octagon = 8
};
```

(c) (3 marks) Write a piece of code that dynamically allocates an array of 10 pointers to `polygonT`.

```cpp
polygonT** list = new polygonT* [10];
```
7. (4 marks) The following segment of code reads integers from the user and computes the product. It stops when the user enters zero.

```cpp
int n;
int p = 1;

cout << "Enter an integer, 0 to end: ";
n = GetInteger();
while (n != 0) {
    p *= n;
    cout << "Enter an integer, 0 to end: ";
n = GetInteger();
}
cout << "The product is: " << p << endl;
```

Solve the above loop-and-half problem by applying the following pattern:

```cpp
int n;
int p = 1;

while (true) {
    cout << "Enter an integer, 0 to end: ";

    n = GetInteger();

    if (n == 0) break;

    p *= n;
}
cout << "The product is: " << p << endl;
```
8. (2 marks) How would you construct an object vec of the Vector class of base type string?

    Vector<string> vec;

9. (3 marks) In reverse Polish notation (RPN), to compute 1.0 + 2.0, you enter the following sequence of keys:

    1.0 ENTER 2.0 +

    Write the sequence of keys you would enter to compute 5.0 * (4.0 + 3.0) / 2.0. Think about it in the content of stack.

    5.0 ENTER 4.0 ENTER 3.0 + * 2.0 /