CS 7A - Spring 2016 - Midterm 2. 5/3/16

Write your responses to following questions on separate paper. Use complete sentences where appropriate and write out code using proper style and syntax. You can hand write your responses or type set with a computer. Due Tuesday, May 10.

1. Consider the following code fragment for the user-defined type, `Token`:

```
struct Token{
    char kind;
    double value;
    string varname;
    Token(char ch) :kind(ch), value(0) { }
    Token(char ch, double val) :kind(ch), value(val) { }
    Token(char ch, string val) :kind(ch), varname(val) { }
};
```

   (a) Describe the member variables of this `struct`.

   (b) What `kind` values are used to declare a variable?

   (c) Describe in detail the ways in which the constructors for `Token` are used.

   (d) Describe how to change `Token` from a `struct` to a `class`.

2. Consider the following code fragment for the user-defined type, `Token_stream`

```
struct Token_stream {
    bool full;
    Token buffer;
    istream & str;
    Token_stream(istream & arg) : str(arg), full(0), buffer(' ') { }
    Token get();
    void unget(Token t) { buffer=t; full=true; }
    void ignore(char);
};
```

   (a) Describe the variable `buffer`. What is it? What is it for? Give an example of how it is used.

   (b) Describe the variable `str`. What is it? What is it for? Give an example of how it is used.

3. Consider the following code fragment for the member function `get()`.

```
Token Token_stream::get()
{
    if (full) { full=false; return buffer; }
    char ch;
    str >> ch;
    if (! str) return(Token(quit));
    switch (ch) {
    case '(':   case ')': case '+': case '-': case '*':
    case '/': case '%': case ';': case '=': case ',':
        return Token(ch);
    case '.': case '0': case '1': case '2': case '3':
    case '4': case '5': case '6': case '7': case '8':
    case '9':
    {   str.unget();
        double val;
        str >> val;
        if (! str) error("Bad token");
        return Token(number,val);
    }
    default:
        if (isalpha(ch) || ch == '_') {
            string s;
            s += ch;
            while(str.get(ch) &&
                 (isalpha(ch) || isdigit(ch) || ch == '_'))
```

```
                    s += ch;
27              str.unget();
                if (! str) error("Bad token");
29              if (s == "let") return Token(let);
                if (s == "const") return Token(constant);
31              if (s == "reset") return Token(reset);
                if (s == "sqrt") return Token(sqroot);
33              if (s == "pow") return Token(power);
                if (s == "help") return Token(help);
35              if (s == "quit" || s == "exit")
                    return Token(quit);
37              return Token(name,s);
            }
39          error("Bad token");
            return Token(' ');
41      }
    }
```

(a) What does `get()` if `full==true`?

(b) What is `str` here?

(c) What is the purpose of `str.unget()` on line 14?

(d) Describe what happens in the `default case`. How does it provide for the declaration of a new variable? How does handle a built-in function like `pow()`? How does it recognise an existing variable in `symbol_table`?

4. Consider the following code fragment for handling `Variable`s in the calculator.

```
struct Variable {
    string name;
    double value;
    bool immutable;
    Variable(string n, double v, bool b) :
        name(n), value(v), immutable(b) { }
};

//  The active variables.
class Symbol_table {
    vector<Variable> names;
public:

double get(string s) {
    for (int i = 0; i<int(names.size()); ++i)
        if (names[i].name == s) return names[i].value;
    error("get: undefined name ",s);
        return 0.0;
}

void set(string s, double d)
{    for (int i = 0; i<=int(names.size()); ++i)
        if (names[i].name == s) {
            names[i].value = d;
            return;
        }
    error("set: undefined name ",s);
}
```

(a) Describe the constructor for a `Variable`. How does it work?

(b) What is `Symbol_table`. What purpose does it serve?

(c) Why does this `get()` function not collide with `Token_stream`'s `get()` function?

(d) Describe in detail how `get()` works and what its purpose is.

(e) Describe in detail how `set()` works and what its purpose is.

(f) Could these `get()` and `set()` functions be made member functions of the `struct Variable`? Discuss.