C++ Reference Sheet - Midterm

Include Headers
```
#include <headerfile>
```

Namespace
```
using namespace std;
using std::cout;
```

Data Types
```
char, bool, short, int, long, unsigned
long, float, double, long double, void
```

Data Types Aliases
```
size_t, uint8_t, uint16_t, uint32_t
```

Data Definitions
```
type var1, var2=value, var3;
type var;
const type var = value;
auto var = value;
```

Literals
```
1.2E-5, 2, 3., 'a', "Hi", true, false,
3.1F, 45L, 0b011, 0xF1, 1'000, R"(.)"
```

Escape sequences
```
\n, \t, \\, \', \"
```

Comments
```
// Comment text
/* Multi-line comment text */
```

Assignment Operator
```
lvar = rvalue;
lvar1 = lvar2 = rvalue;
```

Arithmetic Operators
```
+, -, *, /, %
+=, -=, *=, /=, %/
```

Relational Operators
```
<, <=, >, >=, ==, !=
```

Logical Operators
```
||, &&, !
```

Decisions
```
if ( expression ){
  statements;
} else if ( expression ) {
  statements;
} else {
  statements;
}
```

Loops
```
while (expression) {
  statements;
}
```

I/O Operations
```
cin >> var1, var2;
cout << "TEXT" << var << endl;
getline(cin, inputLine);
cin.get(charVar);
cin.ignore();
```

File I/O
```
ifstream iFile;
ofstream oFile;
xfile.open("file_name");
xfile.open(stringVar));
iFile >> var;
oFile << var;
xfile.close();
```

Formatting Output
```
setw(n), fixed, showpoint,
setprecision(n), left, right
```

Function Call
```
var = fnctName(var1, var2);
```

Function Header and Body
```
type fnctName(type var1, type var2){
  statements;
  return var;
}
```

Other Built-In Functions
```
sizeof(…), static_cast<type>(var)
```

Structures
```
struct StrName {
  type var1 = value;
  type fnctName1(…) {…}
  ...
};
StrName strVar;  // structure variable
var2 = strVar.var1; // member access
strVar.fnctName1(…);
StrName strVar = {arg1, arg2, ...};
```

Pointers
```
type *typePtr;
typePtr = &var1;
*typePtr = var2;
typePr = new type;
delete typePr;
```

C++ Reference Sheet - Midterm

`nullptr;`

STL

string

```
string stringVar;
string stringVar(size, character);
stringVar = "Text";
stringVar.length();
stringVar.back();
stringVar.empty(); // bool
stringVar.erase(position, length);
stringVar.c_str();
stringVar.assign(size, character);
stringVar.substr(position, length);
stringVar.find_first_of(string, pos);
stringVar.find_last_of(string, pos);
string::pos // not found value
stringVar = stringVar1 + stringVar2;
stringVar += stringVar1;
stringVar = to_string(numeric_value);
ivar = stoi(string);
dvar = stod(string);
```

vector

```
vector<type> vectorVar;
vector<type> vectorVar(size, value);
vector<type> vectorVar = vectorVar2;
vector<type> vectorVar = {init_list};
vector<type> vectorVar{init_list};
vectorVar.size();
vectorVar.resize(size, value);
vectorVar.empty(); // bool
vectorVar.clear();
vectorVar.back();
vectorVar.push_back(value);
vectorVar.pop_back(value);
vectorVar[.]
vector< vector<type> > vectorVar
vectorVar[.][.]
```

iostream

```
cout << data; cin >> data;
cin.get(cVar), cVar=cin.get()
cin.ignore(number, character)
getline(cin, stringVar)
```

fstream

```
ofstream, ifstream
ofstream streamObj(fileName);
streamObj.open(fileName);
streamObj.close();
```

iomanip

```
setw(), fixed, setprecision(),
showpoint, left, right, scienfitic,
noshowpoint, defaultfloat
```

cmath

```
abs, cos, exp, fmod, log, log10, sin,
sqrt, pow, tan, atan2, acos, asin
```

random

```
random_object rEngine;
uniform_int_distribution<type>
iDist(first,last);
uniform_float_distribution<type>
fDist(first,last);
iVar = rEngine();
iVar = iDist(rEngine);
```