**CIS 200 - Lab 5**

**Fall 2016**

Consider that individual nodes in an unsorted linked list have the following definition:

|  |
| --- |
| class Vector  {  public:  Vector(int s = 0){ // makes Size = s,  //allocates s space,  // makes all entries 0  }  Vector(const Vector & rhs){ // copy constructor  // makes self a deep copy of rhs  }  Vector operator = (const Vector & rhs){// makes self a deep copy of rhs    }  ~Vector(){ // default destructor    }  int & operator[](int index){ // if 0 <=pos<size  // returns entries[index]    }  int sz(){// returns the # of entries.    }  private:  int size; //store the # of elements used  int \*entries;  };  ostream & operator<<(ostream & out, Vector & rhs); |

# Task 1

Implement all the functions of the above class.

# Task 2

Show your class works by running the following main.

int main(){

Vector test(10);

Vector \*vecPtr;

Vector exam(5);

for (int i = 0; i < test.sz(); i++){

test[i] = i;

}

cout << "test initally is : ";

cout << test << endl;

cout << "empty Vector of length 5 is : ";

cout << exam << endl;

exam = test;

cout << "after exam = test, exam is : ";

cout << exam << endl;

{

Vector sqrs(10);

cout << "The squares are : ";

for (int i = 0; i < sqrs.sz(); i++)

sqrs[i] = i\*i;

vecPtr = &sqrs;

cout << sqrs << endl;

}

cout << "trying to print an empty vector: ";

cout << \*vecPtr << endl;

for (int i = 0; i < test.sz(); i++)

test[i] = 10 - i;

cout << "testing deep copy :\n";

cout << "test is now : ";

cout << test << endl;

cout << "exam is now : ";

cout << exam << endl;

return 0;

}

What to turn in:

A .doc or docx document with your implementation and a screen shot of your output with the testing main.