SUBMITTED BY: BREITENBERGER, NICODEMUS

1 Exercise 1

```
1
2  #include <BALL/FORMAT/PDBFile.h>
3  #include <BALL/KERNEL/system.h>
4  #include <BALL/KERNEL/chain.h>
5  #include <BALL/KERNEL/residue.h>
6  #include <BALL/KERNEL/atom.h>
7  #include <BALL/KERNEL/protein.h>
8
9
10 int main(int argc, char* argv[]){
11
12    BALL::Vector3 translationVector(1,1,1);
13    BALL::System translationSystem;
14    BALL::PDBFile sourceFile;
15    BALL::PDBFile translateFile;
16
17    //Sanity checks for command-line arguments
18    if(argc == 3){
19
20       sourceFile.open(argv[1], std::ios::in);
21       translateFile.open(argv[2], std::ios::out);
22
23    }else if(argc == 6){
24
25    sourceFile.open(argv[1], std::ios::in);
26    translateFile.open(argv[2], std::ios::out);
27
28    translationVector = BALL::Vector3(std::stod(argv[3]), std::stod
      (argv[4]), std::stod(argv[5]));
29
30    }else{
31
32       std::cout << "Wrong amount of Parameters\n\n Useage: prog
      inFile outFile\n";
33       return 1;
34
35    }
36
37    if(sourceFile.is_open()){
38
39       //the read action for a pdbfile reads the data into a system
       which can be manipulated afterwards - and saved
40       sourceFile.read(translationSystem);
```

```
41        sourceFile.close();
42
43        // iterate over all proteins
44        for(BALL::MoleculeIterator m_it = translationSystem.
      beginMolecule(); +m_it; ++m_it){
45
46            //We need to check if the molecule we just grabbed with
      m_it is indeed a protein. if it is not,
47            //We cannot iterate over chains and residues.
48            if (BALL::RTTI::isKindOf<BALL::Protein>(*(m_it))){
49
50                BALL::Protein* protein = BALL::RTTI::castTo<BALL::
      Protein>(*(m_it));
51
52                if(protein->countChains() > 0){
53
54                  for(BALL::ChainIterator ch_it = protein->beginChain
      (); +ch_it; ++ch_it){
55
56                    for(BALL::ResidueIterator r_it = ch_it->
      beginResidue(); +r_it; ++r_it){
57
58                      for(BALL::AtomIterator a_it = r_it->beginAtom
      (); +a_it; ++a_it){
59
60                        //Exercise 1: Translating Atom-Positions
61                        a_it->setPosition(a_it->getPosition() +
      translationVector);
62                      }
63                    }
64                  }
65                }
66            }
67        }
68    }else{
69
70        std::cout << "Could not open PDB file." << '\n';
71
72    }
73
74    //Writing translatedFile
75
76
77    if(translateFile.is_open()){
78
79        translateFile.write(translationSystem);
80        translateFile.close();
81
82    }else{
```

```
83
84          std::cout << "Could not write translated PDB file." << '\n';
85
86      }
87
88      return 0;
89
90  }
```

2 Exercise 2

```
1
2   #include <BALL/FORMAT/PDBFile.h>
3   #include <BALL/KERNEL/system.h>
4   #include <BALL/KERNEL/chain.h>
5   #include <BALL/KERNEL/residue.h>
6   #include <BALL/KERNEL/atom.h>
7   #include <BALL/KERNEL/protein.h>
8
9
10  int main(int argc, char* argv[]){
11
12      BALL::Vector3 translationVector(1,1,1);
13      BALL::System translationSystem;
14      BALL::PDBFile sourceFile;
15      BALL::PDBFile translateFile;
16
17      //Sanity checks for command-line arguments
18      if(argc == 2){
19
20          sourceFile.open(argv[1], std::ios::in);
21
22      }else{
23
24          std::cout << "Wrong amount of Parameters\n\n Useage: prog
        inFile\n";
25          return 1;
26
27      }
28
29      if(sourceFile.is_open()){
30
31          //the read action for a pdbfile reads the data into a system
         which can be manipulated afterwards - and saved
```

```
32        sourceFile.read(translationSystem);
33        sourceFile.close();
34
35        // iterate over all proteins
36        for(BALL::MoleculeIterator m_it = translationSystem.
      beginMolecule(); +m_it; ++m_it){
37
38            //We need to check if the molecule we just grabbed with
      m_it is indeed a protein. if it is not,
39            //We cannot iterate over chains and residues.
40            if (BALL::RTTI::isKindOf<BALL::Protein>(*(m_it))){
41
42                BALL::Protein* protein = BALL::RTTI::castTo<BALL::
      Protein>(*(m_it));
43
44                if(protein->countChains() > 0){
45
46                  for(BALL::ChainIterator ch_it = protein->beginChain
      (); +ch_it; ++ch_it){
47
48                    for(BALL::ResidueIterator r_it = ch_it->
      beginResidue(); +r_it; ++r_it){
49
50                      for(BALL::AtomIterator a_it = r_it->beginAtom
      (); +a_it; ++a_it){
51
52                        //Exercise 2: Printing C-Alphas
53                        if(a_it->getName() == "CA"){
54
55                          std::cout << a_it->getPosition() << '\n
      ';
56
57                        }
58                      }
59                    }
60                  }
61                }
62            }
63        }
64    }else{
65
66        std::cout << "Could not open PDB file." << '\n';
67
68    }
69
70    return 0;
71
72 }
```

3 Exercise 3

3.1 $y''(x) + y(x) = \sin(2x)$

General solution will be from the form $y(x) = y_c(x) + y_p(x)$, which is the sum of the complementary and particular solution.

First determine the complementary solution:

$$y''(x) + y(x) = 0$$

Which can be solved by a sin(x) as well as a cos(x). Because

$$\partial_x^2 \sin(x) = -\sin(x)$$
$$\partial_x^2 \cos(x) = -\cos(x)$$

Therefore

$$y_c(x) = A\sin(x) + B\cos(x)$$

Second step is to determine the particular solution:

$$y_p(x) = C\sin(2x) + D\cos(2x)$$
$$\partial_x^2 y_p(x) = -4C\sin(2x) - 4D\cos(2x)$$

$$-4C\sin(2x) - 4D\cos(2x) + C\sin(2x) + D\cos(2x) = \sin(2x)$$
$$-3C\sin(2x) - 3D\cos(2x) = \sin(2x)$$
$$\Rightarrow D = 0 \quad C = -\frac{1}{3}\sin(2x)$$

$$\boxed{y(x) = A\sin(x) + B\cos(x) - \frac{1}{3}\sin(2x)}$$

Proof:

$$y''(x) = -A\sin(x) - B\cos(x) - \frac{4}{3}\sin(2x)$$

$$y''(x)+y(x) = \sin(x)+B\cos(x)-\frac{1}{3}\sin(2x)-A\sin(x)-B\cos(x)-\frac{4}{3}\sin(2x) = \sin(2x)$$

3.2 $y''(x) + y(x) = \sin(x)$

Complementary solution same as in section 3.1.

$$y_c(x) = A\sin(x) + B\cos(x)$$

Guess for particular solution:

$$y_p(x) = Cx\cos(x)$$

$$y_p'(x) = C\cos(x) - Cx\sin(x)$$

$$y_p''(x) = -C\sin(x) - Cx\cos(x) - C\sin(x) = -2C\sin(x) - Cx\cos(x)$$

$$-2C\sin(x) - Cx\cos(x) + Cx\cos(x) = \sin(x)$$

$$\Rightarrow C = -\frac{1}{2}$$

$$\boxed{y(x) = A\sin(x) + B\cos(x) - \frac{1}{2}x\cos(x)}$$

Proof:

$$y''(x) = -A\sin(x) - B\cos(x) + \sin(x) - \frac{1}{2}x\cos(x)$$

$$y''(x)+y(x) = A\sin(x)+B\cos(x)-\frac{1}{2}x\cos(x)-A\sin(x)-B\cos(x)+\sin(x)-\frac{1}{2}x\cos(x)$$

$$= \sin(x)$$

3.3 $y''(x) - y(x) = 0$

This is a homogeneous differential equation. Therefore it is only necessary to find the homogeneous solution $y_c(x)$.

The equation can be solved by e^x and $e^- x$.

$$\boxed{y(x) = Ae^x + Be^{-x}}$$

Proof:

$$y''(x) = Ae^x + (-1)^2 Be^{-x} = Ae^x + Be^{-x}$$

$$y''(x) - y(x) = Ae^x + Be^{-x} - Ae^x - Be^{-x} = 0$$