```
#include<iostream>
using std::cout;
using std::endl;
#include <sstream>
using std::istringstream;
#include <vector>
using std::vector;

#include<Animal.h>

int main(int argc, char* argv[])
{
  if (argc != 2) {
    cout << "Usage: " << argv[0] << " int" << endl;
    return -1;
  }

  istringstream istr(argv[1]);
  int key;
  istr >> key;
  if (istr.fail()) {
    cout << "Error reading argument: not an integer"
         << endl;
    return -1;
  }

  switch(key) {
  case 0: {
    Quagga q(1888, 1901, true);
  }
    break;
  case 1: {
    Fish f(2014, 0.19, 20, true);
  }
    break;
```

```
  case 2: {
    vector<Mammal> herd;
    herd.push_back(Quagga(1880, 1882, true));
    herd.push_back(Quagga(1880, 1890));
  }
    break;
  case 3: {
    vector<Mammal*> herd;
    herd.push_back(new Quagga(1880, 1882));
    herd.push_back(new Quagga(1880, 1890));
  }
    break;
  case 4: {
    Quagga* ptr = new Quagga(1879, 1880, true);
    if (dynamic_cast<Mammal*>(ptr)) delete
dynamic_cast<Mammal*>(ptr);
  }
    break;
  case 5: {
   Mammal* ptr = new Mammal(1879, true);
    if (dynamic_cast<Quagga*>(ptr)) delete
dynamic_cast<Quagga*>(ptr);
  }
    break;
  case 6: {
    vector<Animal*> animals;
    animals.push_back(new Fish(2014, 0.99, 10));
    animals.push_back(new Quagga(1883, 1888));
    animals.push_back(new Mammal(2013));
    animals.push_back(new Animal(2012));
    for(unsigned int i=0; i < 4; i++) {
      cout << "Index = " << i << " Number = "
           << animals[i]->Number() << " Warm up method = ";
      animals[i]->WarmUp();
    }
  }
```

```
    break;
  default : cout << "Unrecognized key" << endl;
  }

  return 0;
}
```