Lab 06
Inheritance and Polymorphism

Section 1: Guess program outputs.

1. What will the following program display?

```
#include <iostream>
#include <memory>
using namespace std;

class First
{
protected:
  int a;
public:
  First (int x = 1) { a = x; }
  int getVal() const { return a; }
};

class Second : public First
{
private:
  int b;
public:
  Second(int y = 5) { b = y; }
  int getVal() const { return b; }
};

int main()
{
    shared_ptr<First> object1 = make_shared<First>();
    shared_ptr<Second> object2 = make_shared<Second>();
    cout << object1->getVal() << endl;
    cout << object2->getVal() << endl;
    return 0;
}
```

2. What will the following program display?

```
#include <iostream>
#include <memory>
using namespace std;

class First
{
protected:
  int a;
public:
  First(int x = 1) { a = x; }
  void twist() { a *= 2; }
  int getVal() { twist(); return a; }
};

class Second : public First
{
private:
  int b;
public:
  Second(int y = 5) { b = y; }
  void twist() { b *= 10; }
};

int main()
{
    shared_ptr<First> object1 = make_shared<First>();
    shared_ptr<Second> object2 = make_shared<Second>();
    cout << object1->getVal() << endl;
    cout << object2->getVal() << endl;

    return 0;
}
```

Section 2: Review Questions and Exercises

Suppose that the classes Dog and Cat derive from Animal, which in turn derives from Creature. Suppose further that pDog, pCat, pAnimal, and pCreature are pointers to the respective classes. Suppose that Animal and Creature are both abstract classes.

1. Will the statement
Animal a;
compile?

2. Will the statement
pAnimal = new Cat;
compile?

3. Will the statement
pCreature = new Dog;
compile?

4. Will the statement
pCat = new Animal;
compile?

5. Rewrite the following two statements to get them to compile correctly.
pAnimal = new Dog;
pDog = pAnimal;

Section 3: Programming Challenges

1. Sequence Sum

A sequence of integers such as 1, 3, 5, 7, ... can be represented by a function that takes a non-negative integer as parameter and returns the corresponding term of the sequence.
For example, the sequence of odd numbers just cited can be represented by the function body
{ return 2*k – 1; }
Write an abstract class AbstractSeq that has a pure virtual member function
virtual int fun (int k) = 0;
as a stand-in for an actual sequence, and two member functions
void printSeq(int k, int m);
int sumSeq(int k, int m);
that are passed two integer parameters k and m, where k < m. The function printSeq will print all the terms fun(k) through fun(m) of the sequence, and likewise, the function sumSeq will return the sum of those terms. Demonstrate your AbstractSeq class by creating two subclasses that you use to sum the terms of two different sequences which are odd sequence and square sequence, you may create more if you want to. Determine what kind of output best shows off the operation of these classes, and write a program that produces that kind of output.

2. Flexible Encryption

Define a subclass `SimpleEncryption` to complete the encryption program below whose transform function uses an integer key to transform the character passed to it. The function transforms the character by adding the key to it. The key is represented as a member of the Encryption class, and the class has a member function that sets the encryption key.

```
#include <iostream>
#include <fstream>
using namespace std;

class Encryption
{
 protected:
   ifstream inFile;
   ofstream outFile;
   int key;
 public:
   Encryption(char* inFileName, char* outFileName);
   ~Encryption();

   virtual char transform(char ch) = 0;

   void encrypt();
   void setKey(int key){ this->key = key;}
};

Encryption::Encryption(char* inFileName, char* outFileName)
{
   inFile.open(inFileName);
   outFile.open(outFileName);
   if (!inFile)
   {
       cout <<  "The file " << inFileName << " cannot be opened.";
       exit(1);
   }
   if (!outFile)
   {
      cout << "The file " << outFileName << " cannot be opened.";
```

```
       exit(1);
    }
}

Encryption::~Encryption()
{
   inFile.close();
   outFile.close();
}

void Encryption::encrypt()
{
  char ch;
  char transCh;

  inFile.get(ch);
  while (!inFile.fail())
    {
      transCh = transform(ch);
      outFile.put(transCh);
      inFile.get(ch);
    }
}

int main()
{
  char fileIn[] = "fileIn.txt";
  char fileOut[] = "fileOut.txt";

  SimpleEncryption obfuscate(fileIn, fileOut);

  // Get and set the key to be used for the encryption
  cout << "Enter a small integer as an encryption key -> ";
  int key;
  cin >> key;
  obfuscate.setKey(key);

  // Perform the encryption
  obfuscate.encrypt();

  return 0;
}
```