CS 32 Worksheet 2

This worksheet is entirely **optional**, and meant for extra practice. Some problems will be more challenging than others and are designed to have you apply your knowledge beyond the examples presented in lecture, discussion or projects. All exams will be done on paper, so it is in your best interest to practice these problems by hand and not rely on a compiler.

Solutions are written in red. The solutions for **programming** problems are not absolute, it is okay if your code looks different; this is just one way to solve the specific problem.

If you have any questions or concerns please email arabellekezia@ucla.edu or brendon1097@gmail.com, or go to any of the LA office hours.

Concepts

Inheritance and Polymorphism

1.) What changes do you have to make to the following program to have it output “I’m Gene“?

HINT: You will need to use the virtual keyword!

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

class LivingThing {
  public:
     virtual void intro() { cout << "I’m a living thing" <<
endl; }
};

class Person : public LivingThing {
  public:
       // repeating the word virtual is not required here (but
       // recommended as a reminder to a human reader)
     void intro() { cout << "I’m a person" << endl; }
};

class UniversityAdministrator : public Person {
  public:
       // repeating the word virtual is not required here (but
       // recommended as a reminder to a human reader)
     void intro() {
          cout << "I'm a university administrator" << endl;
     }
};

class Chancellor : public UniversityAdministrator {
  public:
       // repeating the word virtual is not required here (but
       // recommended as a reminder to a human reader)
     void intro() { cout << "I'm Gene" << endl; }
};

int main() {
     LivingThing* thing = new Chancellor();
     thing->intro();
     ...
}
```

Difficulty: Easy

2.) Given the following class declarations, complete the implementation of each constructor so that the program compiles. Your implementations should correctly assign constructor arguments to class member variables.

HINT: You will need to use initializer lists!

```
class Animal {
public:
     Animal(string name);
private:
     string m_name;
};

class Cat : public Animal {
public:
     Cat(string name, int amountOfYarn);
private:
     int m_amountOfYarn;
};

class Himalayan : public Cat {
public:
    Himalayan(string name, int amountOfYarn);
};

class Siamese: public Cat {
public:
    Siamese(string name, int amountOfYarn, string toyName);
private:
    string m_toyName;
};

Animal::Animal(string name)
    : m_name(name) {}

Cat::Cat(string name, int amountOfYarn)
    : Animal(name), m_amountOfYarn(amountOfYarn) {}

Himalayan::Himalayan(string name, int amountOfYarn)
    : Cat(name, amountOfYarn) {}

Siamese::Siamese(string name, int amountOfYarn, string toyName)
    : Cat(name, amountOfYarn), m_toyName(toyName) {}
```

Difficulty: Easy

3.) Would the following work in C++? Why or why not?

```
class B;
class A : public B {// Code for A};
class B : public A {// Code for B};
```

Conceptually, this code is saying "A is a proper subset of B, and B is a proper subset of A", which is nonsense.

Practically, every object of a derived class contains an instance of the base class. If the code above were legal, a B object would contain an A object that contains a B object that contains an A object, ad infinitum.

Difficulty: Medium

4.) What is the output of the following program?

```
class Pet {
public:
        Pet() { cout << "Pet" << endl; }
        ~Pet() { cout << "~Pet" << endl; }
};

   // This is an unusual class that derives from Pet but also
   // contains a Pet as a data member.
class Dog : public Pet {
public:
        Dog() { cout << "Woof" << endl; }
        ~Dog() { cout << "Dog ran away!" << endl; }
private:
        Pet buddy;
};

int main() {
        Pet* milo = new Dog;
        delete milo;
}
```

```
Pet
Pet
Woof
~Pet
```

Undefined behavior after this, because Pet's destructor is not declared virtual.

Difficulty: Medium

5.) Suppose the class declaration for Pet was changed as shown below. What is the new output of the code in problem 4) with these new changes?

```
class Pet {
public:
        Pet() { cout << "Pet" << endl; }
        virtual ~Pet() { cout << "~Pet" << endl; }
};
```

```
Pet
Pet
Woof
```

```
Dog ran away!
~Pet
~Pet
```

Difficulty: Medium

6.) The following code has several errors. Rewrite the code so that it can successfully compile. Try to catch the errors without using a compiler!

```
class LivingThing {
public:
     LivingThing(int a) { age = a; }
     void myBirthday() { age++; }
private:
     int age;
};

class Person : public LivingThing {
public:
     Person(int a) : LivingThing(a) { ~~age = a;~~ }
     void birthday() {
          ~~age++;~~
          myBirthday();
     }
};
```

Difficulty: Easy

7.) Examine the following code and determine its output. (JKC)

```
#include <iostream>
#include <string>

using namespace std;

class A {
public:
     A() : m_val(0) {
          cout << "What a wonderful world! " << m_val << endl;
     }
     virtual ~A() { cout << "Guess this is goodbye " << endl; }
     virtual void saySomething() = 0;
     virtual int giveMeSomething() = 0;
```

```
private:
      int m_val;
};

class B : public A {
public:
      B() : m_str("me"), m_val(1) {
            cout << m_str << " has just been birthed."  << endl;
      }
      B(string str, int val) : m_str(str), m_val(val) {
            cout << "More complex birth " << m_str << endl;
      }
      ~B() {
            cout << "Why do I have to leave this world!" << endl;
      }
      virtual void saySomething() {
            cout << "Coming in from " << m_str << " with "
                 << giveMeSomething() << endl;
      }
      virtual int giveMeSomething() { return m_val*5; }
private:
      int m_val;
      string m_str;
};

class C {
public:
      C() : m_val(2) {
            m_b = new B("C", m_val);
            cout << "Hello World!!" << endl;
      }
      C(B b, int val) :  m_val(val) {
            m_b = new B(b);
            cout << m_b->giveMeSomething() << endl;
      }
      ~C() {
            m_b->saySomething();
            delete m_b;
            cout << "Goodbye world!" << endl;
      }
private:
      B* m_b;
      int m_val;
```

```
};

int main() {
	B* b_arr = new B[5];
	for(int i = 0; i < 5; i++) {
		b_arr[i].saySomething();
	}
	B b("B", 5);
	A* a = &b;
	cout << a->giveMeSomething() << endl;
	C c;
	C c2(b, b.giveMeSomething());
	delete [] b_arr;
}
```

Difficulty: Hard

```
What a wonderful world! 0
me has just been birthed.
What a wonderful world! 0
me has just been birthed.
What a wonderful world! 0
me has just been birthed.
What a wonderful world! 0
me has just been birthed.
What a wonderful world! 0
me has just been birthed.
Coming in from me with 5
Coming in from me with 5
Coming in from me with 5
Coming in from me with 5
Coming in from me with 5
What a wonderful world! 0
More complex birth B
25
What a wonderful world! 0
More complex birth C
Hello World!!
25
Why do I have to leave this world!
Guess this is goodbye
Why do I have to leave this world!
Guess this is goodbye
```

```
Why do I have to leave this world!
Guess this is goodbye
Why do I have to leave this world!
Guess this is goodbye
Why do I have to leave this world!
Guess this is goodbye
Why do I have to leave this world!
Guess this is goodbye
Coming in from B with 25
Why do I have to leave this world!
Guess this is goodbye
Goodbye world!
Coming in from C with 10
Why do I have to leave this world!
Guess this is goodbye
Goodbye world!
Why do I have to leave this world!
Guess this is goodbye
```