ECE326 – Fall 2019: Week 3 Exercise Questions

1. True or False **[1 mark each]**

Circle T is true, otherwise circle F for false.

1. Overloading virtual functions in C++ is an example of multiple dispatch. **T** **(F)**

   Function overloading in C++ is done statically (static dispatch), at compile time.

2. You can declare an opaque data type on the stack. **T** **(F)**

   No. You must know the size of a data type to be able to declare it on the stack, but an opaque data type is not defined, making it impossible to calculate its size.

3. Pure virtual functions are not necessarily pure functions. **(T)** **F**

   Pure virtual functions can still have side effects such as modifying global variables, which makes it not always a pure function.

4. A virtual function overloading an operator is an example of dynamic dispatch. **(T)** **F**

   Yes, for example, `virtual int operator+(int val) const;`

5. Dynamically-typed interpreted language cannot implement early binding. **(T)** **F**

   Only compiled languages can legitimately implement early binding (name to address translation)

2. Short Answers

1. Override the eat method in Animal so that the eat method in Dog will, *in addition to what* Animal.eat *already does*, print "Wags its tail" at the very end. Show the entire class definition for Dog. **[3 marks]**

```
class Animal:
    …
    # may change this function in the future
    def eat(self, food):     # a mistake was fixed (self added)
        print(str(food) + " is delicious")


class Dog(Animal):
    def eat(self, food):
        super().eat(food)        # or Animal.eat(self, food)
        print("Wags its tail")
```

2. Write an expression with equivalent result as the following list comprehension, using only the map and filter function for control flow. **[2 marks]**

```
[ str(v) for v in range(10, 100) if not (v//10+v%10)%7 ]
```

```
def cond(v):
    return not (v//10+v%10)%7

b = list(map(str, filter(cond, range(10, 100))))
```

3. Prototype-based Programming **[10 marks]**

In Prototype-based programming, all objects have the type Object. The base object initially has no attribute. We wish to program with this paradigm in Python. Create a Person object out of the base object, followed by inheriting from the Person object to create a Student object. Finally, make an instance of Student. A Person object should have the data attributes: name, age, and gender, with a method called birthday() that increments age. A Student object should have the data attributes: id, gpa, and year. Create an instance of Student named "me" with your credential (can be fake). Choose suitable defaults for the prototypes.

```
class Object:
     pass

base = Object()
```

```
import copy
person = copy.deepycopy(base)
person.name = ""
person.age = 26
person.gender = "unknown"
def bday(self):
    self.age += 1
Object.birthday = bday
student =
copy.deepycopy(person)
```

```
student.id = 0
student.gpa = 0
student.year = 1900
me = copy.deepcopy(student)
me.name = "jack"
me.gender = "male"
me.id = 123456789
me.gpa = 3
me.year = 2015
```

4. Virtual Tables **[10 marks]**

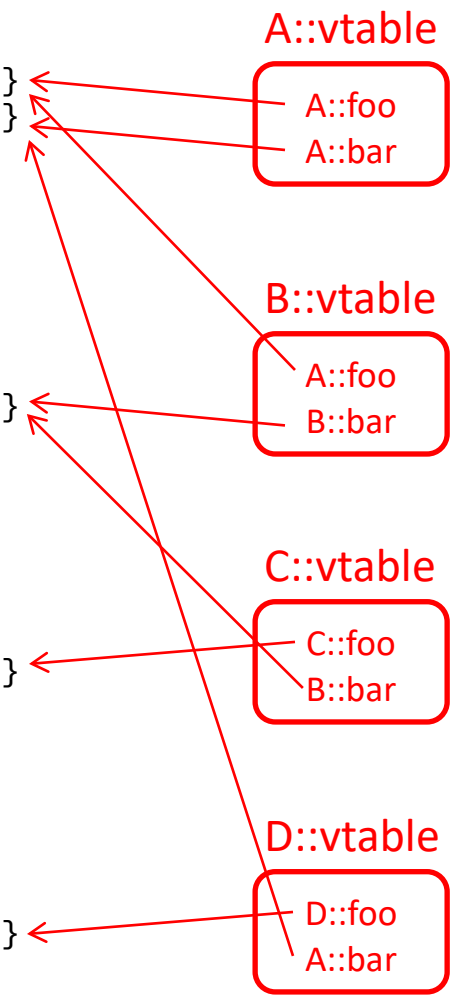
a. For the following inheritance hierarchy, draw a virtual table for each class and draw an arrow from each entry in the virtual table to their definition in the C++ classes. [7 marks]

```
struct A {
  virtual void foo() { cout << “A.foo”; }
  virtual void bar() { cout << “A.bar”; }
  void baz() { cout << “A.baz”; }
};

struct B : public A {
  virtual void bar() { cout << “B.bar”; }
};

struct C : public B {
  virtual void foo() { cout << “C.foo”; }
  void baz() { cout << “C.baz”; }
};

struct D : public A {
  virtual void foo() { cout << “D.foo”; }
  void baz() { cout << “D.baz”; }
};
```



b. What is the output of the following program? [3 marks]

```
D d = D();
C c = C();
A * ad = &d;
A * ac = &c;

ac->baz();
ad->foo();
ac->bar();
```

```
A.baz
D.foo
B.bar
```