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**
2. You can declare an opaque data type on the stack. **T** **F**
3. Pure virtual functions are not necessarily pure functions. **T** **F**
4. A virtual function overloading an operator is an example of dynamic dispatch. **T** **F**
5. Dynamically-typed interpreted language cannot implement early binding. **T** **F**

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(food):
        print(str(food) + " is delicious")
```

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 ]
```

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()
```

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();
```