EECS2030 Fall 2017 **Name: (Last, First)** _____
Advanced OO Programming
Sample Exam Questions Solution

---

1. Consider the following classes, where we use `print` to abbreviate `System.out.println`:

```
interface I {
  void mi();
}
```

```
class A implements I {
  void mi() {
    println("A.mi"); }
}
```

```
class B implements I {
  void mi() {
    println("B.mi"); }
}
```

```
1  class Collector {
2    A[] as; int numberOfAs;
3    B[] bs; int numberOfBs;
4    Collector() {
5      as = new A[10]; bs = new B[10]; }
6    void addA(A a) {
7      as[numberOfAs] = a; numberOfAs++; }
8    void addB(B b) {
9      bs[numberOfBs] = b; numberOfBs++; }
10   void callAll() {
11     for(int i = 0; i < numberOfAs; i ++)
12     { as[i].mi(); }
13     for(int i = 0; i < numberOfBs; i ++)
14     { bs[i].mi(); }
15   }
16 }
```

```
1  class Tester {
2    static void main(String[] args) {
3      I i = new I();
4      B b = new B(); A a = new A();
5      Collector c = new Collector();
6      c.addB(b); c.addA(a);
7      c.callAll();
8    }
9  }
```

(a) Explain if the assignment `as[numberOfAs] = a` in **Line 7** of the above `Collector` class compiles.

> **Solution:**
> Yes, because `a`'s static type `A` is a descendant class of `as[i]`'s static type `A`.

[of 5 marks]

(b) Explain if the method call `as[i].mi()` in **Line 12** of the above `Collector` class compiles.

> **Solution:**
> Yes, because `as[i]`'s static type `A` has the method `mi` defined.

[of 5 marks]

(c) Explain if **Line 3** of the above `Tester` class compiles.

> **Solution:**
> No, because `I` being an interface cannot be used as a dynamic type.

[of 5 marks]

(d) Write and Explain the console output from **Line 7** of the above `Tester` class.

> **Solution:**
> Output is:

```
A.mi
B.mi
```

- The first `for` loop in method `callAll` will call the version of method `mi` implemented in class `A`. So the output is: `"A.mi"`.
- The second `for` loop in method `callAll` will call the version of method `mi` implemented in class `B`. So the output is: `"B.mi"`.

[of 10 marks]

(e) The above `Collector` class does not make use of *polymorphism*, which results from the fact that classes `A` and `B` implement a common interface `I`. Rewrite the above `Collector` class, such that there is only one array attribute and one `add` method, and that the `callAll` method contains just a single loop.

Solution:

```
1  class Collector {
2    I[] is; int numberOfIs;
3    Collector() {
4      is = (I[]) new Object[10]; }
5    void addI(I i) {
6      is[numberOfIs] = i; numberOfIs++; }
7    void callAll() {
8      for(int i = 0; i < numberOfIs; i ++)
9      { is[i].mi(); }
10   }
11 }
```

[of 15 marks]