

# SE1021 Half Exam 1 Name:

---

Use only your pen/pencil/eraser (for example, no note-sheet). Review all questions before you get started. The exam is printed double-sided. Show all work. You do not need to comment your code.

1. (10 points) **Describe** the difference in behavior between `x++` and `++x`.
  
2. (10 points) **Write** what the following code-snippet prints when it is run:  

```
char c1 = 'b';  
char c2 = (char)(c1 + 2);  
System.out.println("c2: "+c2);
```
  
3. (10 points) **Edit** the first line of this if-statement to not crash even if `str` is `null`. The code should only print "Hi Evan" if `str` actually equals "Evan." **Don't add** any new if statements if you can help it.  

```
if(                str.equals("Evan")                ) {  
    System.out.println("Hi Evan");  
}
```
  
4. (10 points) **Describe** two differences between a constructor and an ordinary method.
  
5. (5 points) **Describe** two ways the `this` keyword could be used in a class.
  
6. (5 points) **Write** these types **in order** from least memory to most memory: `int`, `short`, `byte`, `long`.

7. (30 pts.) Typecasting errors are the only errors in this problem. (You may assume the appropriate includes have been done.) Write the types of the left and right hand side of the assignment **that is in bold**. For non-primitive types, write "ref" for a reference. Some of the classes are from the UML diagram on the previous page. The two rows are filled in as an example.

	Code Snippet	Type of left-hand-side	Type of right-hand-side	Will it compile?
(ex)	float f = 4.0f; <b>double d = f;</b>	double	float	yes
a	<b>int i = 5.0;</b>			
b	String str = "asdf"; <b>char c = str;</b>			
c	<b>byte b = (byte)100;</b>			
d	double d = 3.14; <b>float f = d;</b>			

In the next part, assume that Speaker, Dog, and Cat are implemented exactly as follows (not as in class):

```
public interface Speaker { public void speak();}
public class Dog { public void speak() {System.out.println("Woof");}}
public class Cat implements Speaker { public void speak() {System.out.println("Meow");}}
```

(ex)	<b>String str = new Dog();</b>	String ref	Dog ref	no
C	<b>Cat c = new Cat();</b>			
D	<b>Speaker s = new Dog();</b>			
E	<b>Speaker s = new Cat();</b>			
F	Cat c = new Cat(); <b>Speaker s = c;</b>			

8. (20 points) For this problem, assume Person stores name in an instance variable and provides a standard constructor and mutators for it. For partial credit, sketch out what memory will look like when this program runs, using whatever format you desire.

```
Person joe = new Person("Joe");
Person mary = new Person("Mary");
List<Person> people = new ArrayList<>();
people.add(joe);
people.add(mary);
people.get(0).setName("Bob");
System.out.println("joe: "+joe.getName());
System.out.println("mary: "+mary.getName());
System.out.println("p0: "+people.get(0).getName());
System.out.println("p1: "+people.get(1).getName());
```

**Complete** what this program will print when run:

joe:  
mary:  
p0:  
p1: