CMSC 206: Data Structures
Casting and `instanceof`

The exercises refer to the following classes:

```
public class Employee { … }
public class Teacher extends Employee { … }
public class Doctor extends Employee { … }
public class CompSciTeacher extends Teacher { … }
```

Each class is equipped with a default constructor (one that doesn't take any parameters).

For each labeled line below, does that line compile? If not, would it compile if there were a cast? If it does compile (or would with a cast), would it throw an exception? (Consider all the lines as belonging to one method, executing in order from top to bottom. If a line would not compile or would throw an exception, pretend that the program skips that line.)

```
Employee emp;
Teacher teach;
Doctor doc;
CompSciTeacher cst;
```

a. `emp = new Employee();`
b. `emp = new Teacher();`
c. `emp = new CompSciTeacher();`
d. `teach = new CompSciTeacher();`
e. `teach = new Doctor();`
f. `teach = new Employee();`
g. `cst = new CompSciTeacher();`
h. `cst = new Teacher();`
i. `cst = new Employee();`
j. `doc = new Doctor();`
k. `doc = new CompSciTeacher();`
l. `doc = new Employee();`
   `emp = new Teacher();`
m. `teach = emp;`
   `teach = new Teacher();`
n. `emp = teach;`
o. `cst = teach;`
   `cst = new CompSciTeacher();`
p. `teach = cst;`
q. `emp = cst;`
r. `teach = emp;`
s. `cst = emp;`
t. `doc = emp;`

For each labeled chunk below, state whether the `instanceof` would yield `true` or `false`. All of these lines compile and run without error.

```
Employee emp = new Employee();
Teacher teach = new Teacher();
Doctor doc = new Doctor();
CompSciTeacher cst = new CompSciTeacher();
```

a. `emp instanceof Employee`
b. `teach instanceof Employee`
c. `doc instanceof Employee`
d. `cst instanceof Employee`
e. `emp instanceof Teacher`
f. `teach instanceof Teacher`
g. `doc instanceof Teacher`
h. `cst instanceof Teacher`
i. `emp instanceof CompSciTeacher`
j. `teach instanceof CompSciTeacher`
k. `doc instanceof CompSciTeacher`
l. `cst instanceof CompSciTeacher`

```
emp = new Teacher();
```

m. `emp instanceof Employee`
n. `emp instanceof Teacher`
o. `emp instanceof CompSciTeacher`

```
emp = new CompSciTeacher();
```

p. `emp instanceof Employee`
q. `emp instanceof Teacher`
r. `emp instanceof CompSciTeacher`

```
teach = new CompSciTeacher();
```

s. `teach instanceof Employee`
t. `teach instanceof Teacher`
u. `teach instanceof CompSciTeacher`