

Fourth G51PRG self-test

The definitions below will be used as a running example throughout the test.

```
interface FormFiller {  
    public String firstName();  
    public String lastName();  
    public String dateOfBirth();  
    public String address();  
}  
  
class SimpleFormFiller implements FormFiller {  
  
    private String firstName;  
    private String lastName;  
    private String dateOfBirth;  
    private String address;  
  
    public SimpleFormFiller(String f, String l, String d, String a){  
        firstName = new String(f);  
        lastName = new String(l);  
        dateOfBirth = new String(d);  
        address = new String(a);  
    }  
  
    public String firstName() {return firstName;}  
    public String lastName() {return lastName;}  
    public String dateOfBirth() {return dateOfBirth;}  
    public String address() {return address;}  
    public void printForm(){  
        System.out.println(this.firstName());  
        System.out.println(this.lastName());  
        System.out.println(this.dateOfBirth());  
        System.out.println(this.address());  
    }  
}
```

(continued on the next page)

```

class Employee extends SimpleFormFiller {

    public static String companyName = "BigCompany";
    private String payroll;

    public Employee(String f, String l, String d, String a, String p){
        // fill in yourself: see question 1

    }

    public String payroll() {return payroll;}

    public void printForm(){
        super.printForm();
        System.out.println(this.payroll());
    }
}

```

1. Write a constructor for the Employee class which sets all the instance fields to the given parameters. Use the constructor of the superclass to set the first four fields.
2. Does Employee class have to provide an implementation for the methods `firstName()`, `lastName()`, `dateOfBirth()`, and `address()`?

In what follows, assume that the class definitions are correct.

3. Suppose that the following main() function has been added to Employee.
Would it compile?
If yes, what will be printed after it is executed?

```

public static void main(String[] args){
    Employee tom = new Employee("Tom", "Smith", "1980", "Unknown", "1111");
    Employee bob = new Employee("Bob", "Jones", "1970", "London", "1112");
    tom.companyName = "NewCompany";
    System.out.println(bob.companyName);
}

```

4. Can we invoke `dateOfBirth()` method on `tom` (declared as above)?
5. In the main() from question 3, suppose `tom` was declared as a `FormFiller`:

```
FormFiller tom = new Employee("Tom", "Smith", "1980", "Unknown", "1111");
```

Would this line (above) cause a compiler error?

6. Tick all the lines which would cause a compiler error:

```
public static void main(String[] args){  
    SimpleFormFiller tom = new Employee("Tom", "Smith", "1980", "X", "1111");  
    Employee bob = new Employee("Bob", "Jones", "1970", "London", "1112");  
    bob.companyName = "Bob's Company";  
    tom.companyName = "NewCompany";  
}
```

7. Would the compiler report an error for the main() below?

If not, would Tom's payroll number be printed?

```
public static void main(String[] args){  
    SimpleFormFiller tom = new Employee("Tom", "Smith", "1980", "X", "1111");  
    tom.printForm();  
}
```

8. Would the compiler report an error for the main() below?

```
public static void main(String[] args){  
    SimpleFormFiller tom = new Employee("Tom", "Smith", "1980", "X", "1111");  
    tom.payroll();  
}
```

9. Consider the following definitions:

```
interface MyInterface {  
    public Object getData();  
}  
  
class MyClass implements MyInterface {  
    String name;  
    public String getData() { return name; }  
}
```

Would it compile?

10. Write the implementation for `hasGreaterVolume()` which returns `true` if the current object has strictly greater volume, and `false` if it has the same volume or less.

```
public abstract class Vessel {  
    public abstract int getVolume();  
    public boolean hasGreaterVolume(Vessel v){  
    }  
}
```