

Homework 4 Part I test cases (Total: 40 pts)

PersonDB's averageAge I (1 point)

```
Person [] ptest = null;
PersonDB db = new PersonDB(ptest);
db.getAverageAge() -> 0.0;
```

PersonDB's averageAge II (3 pts)

```
Person [] ptest = new Person []{new Person("test1",5), new Person("test2", 16), new
Person("test3", 12)};
PersonDB db = new PersonDB(ptest);
db.getAverageAge() -> 11.0
```

PersonDB's averageAge III (3pts)

```
Person [] ptest = new Person []{null, new Person("test2", 16), new Person("test3", 12)};
PersonDB db = new PersonDB(ptest);
db.getAverageAge() ->14.0
```

PersonDB's averageAge IV (3 pts)

```
Person [] ptest2 = new Person []{new Person("test2", 16),new Person("test3", 12), null};
PersonDB db = new PersonDB(ptest2);
db.getAverageAge() -> 14.0
```

PersonDB's getYoungest I (3 pts)

```
Person [] ptest = new Person []{null, new Person("test2", 16), new Person("test3", 12)};
PersonDB db = new PersonDB(ptest);
db.getYoungest().getAge() -> 12
```

PersonDB's getYoungest II (3 pts)

```
Person [] ptest2 = new Person []{new Person("test3", 3),null, new Person("test3", 12)};
PersonDB db = new PersonDB(ptest2);
db.getYoungest().getAge() -> 3
```

PersonDB's getYoungest III (3 pts)

```
Person [] ptest3 = new Person []{new Person("test3", 3), new Person("test3", 12),null};
PersonDB db = new PersonDB(ptest3);
db.getYoungest().getAge() -> 3
```

PersonDB's getOldest I(3 pts)

```
Person [] ptest = new Person []{null, new Person("test2", 16), new Person("test3", 12)};
PersonDB db = new PersonDB(ptest);
db.getOldest().getAge() -> 16
```

PersonDB's getOldest II (3 pts)

```
Person [] ptest2 = new Person []{new Person("test3", 3),null, new Person("test3", 12)};
PersonDB db = new PersonDB(ptest2);
db.getOldest().getAge() -> 12;
```

PersonDB's getOldest III(3 pts)

Note: In the test output file this test is displayed as PersonDB's getYoungest III but it actually is testing oldest method.

```
Person [] ptest3 = new Person []{new Person("test3", 3), new Person("test3", 12),null};
PersonDB db = new PersonDB(ptest3);
db.getOldest().getAge() -> 12
```

PersonDB's getYoungest & getOldest on empty array I (3 pts)

```
Person [] ptest4 = new Person []{null, null, null};
PersonDB db = new PersonDB(ptest4);
db.getOldest() -> null
getYoungest() -> null;
```

PersonDB's getYoungest & getOldest on empty array II (3 pts)

```
Person [] ptest5 = null;
PersonDB db = new PersonDB(ptest5);
db.getOldest() -> null
db.getYoungest() -> null
```

PersonDB's isInDataBase I(3 pts)

```
Person [] ptest6 = new Person []{new Person("test1", 16), new Person("test2", 16),
new Person("test3", 12)};
PersonDB db = new PersonDB(ptest6);
db.isInDatabase("test3") -> true
db.isInDatabase("test2") -> true
```

PersonDB's isInDataBase II (3 pts)

```
Person [] ptest6 = new Person []{new Person("test1", 16), new Person("test2", 16),
new Person("test3", 12)};
PersonDB db = new PersonDB(ptest6);
db.isInDatabase("test4") -> false
db.isInDatabase("test") -> false
```

Homework 4 Part II (20 points)

Deductions

- 5 points for not catching non-integer inputs
- 1 point for not prompting user to guess initially

Bonus Points

- 5 points for going above and beyond the requirements