CS 251
Intermediate Programming
Quiz 4

Brooke Chenoweth

University of New Mexico

Spring 2025

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(a) `methodOne`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(a) `methodOne`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(b) `methodTwo`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(b) `methodTwo`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(c) `methodThree`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(c) `methodThree`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(d) `methodFour`?

A overrides

B overloads

C hides

Question 1 – Inheritance

```
public class ClassA {
  public void methodOne(int i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodOne(float i) {
  }
  public void methodTwo(int i) {
  }
  public static void methodThree(int i) {
  }
  public static void methodFour(float i) {
  }
}
```

(d) `methodFour`?

A overrides

B overloads

C hides

More Inheritance

```
public class ClassA {
  public void methodFive(int i) {
  }
  public static void methodSix(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public static void methodFive(int i) {
  }
  public void methodSix(int i) {
  }
}
```

What happens here?

More Inheritance

```
public class ClassA {
  public void methodFive(int i) {
  }
  public static void methodSix(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public static void methodFive(int i) {
  }
  public void methodSix(int i) {
  }
}
```

What happens here? Compile-time error! Cannot change instance method to class method, and vice-versa.

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodSeven`?

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodSeven`?

Okay to allow more access than parent.

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodEight`?

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodEight`?

Can't make less accessible than parent.

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodNine`?

More Inheritance

```
public class ClassA {
  protected void methodSeven(int i) {
  }
  protected void methodEight(int i) {
  }
  private void methodNine(int i) {
  }
}
```

```
public class ClassB extends ClassA {
  public void methodSeven(int i) {
  }
  private void methodEight(int i) {
  }
  public void methodNine(int i) {
  }
}
```

`methodNine`?

Private method not visible to subclass, so not overriding here.

Question 2 – Access Modifiers

Ordered from least to most visibility:

Question 2 – Access Modifiers

Ordered from least to most visibility:

`private` Only this class

Question 2 – Access Modifiers

Ordered from least to most visibility:

`private` Only this class

package-private No modifier. This class and others in same package.

Question 2 – Access Modifiers

Ordered from least to most visibility:

`private` Only this class

package-private No modifier. This class and others in same package.

`protected` This class, those in the same package, and subclasses of this class

Question 2 – Access Modifiers

Ordered from least to most visibility:

`private` Only this class

package-private No modifier. This class and others in same package.

`protected` This class, those in the same package, and subclasses of this class

`public` Accessible to all

Question 3 – Keywords

Which are *not* Java keywords?

- bool
- break
- enum
- import
- include
- int
- switch
- if
- then
- else

Question 3 – Keywords

Which are *not* Java keywords?

- bool
- break
- enum
- import
- include
- int
- switch
- if
- then
- else

Question 4 – Compile error

```
public class MyClass {
    private int x = 10;

    public static void main(String[] args) {
        if(x > 5) {
            System.out.println("Big");
        } else {
            System.out.println("Small");
        }
    }
}
```

A Cannot access private variable x from a public method.

B Cannot access x without an instance of MyClass.

C Variable x is out of scope in the main method.

D The println method expects a String, not an int.

Question 4 – Compile error

```
public class MyClass {
    private int x = 10;

    public static void main(String[] args) {
        if(x > 5) {
            System.out.println("Big");
        } else {
            System.out.println("Small");
        }
    }
}
```

A Cannot access private variable x from a public method.

B Cannot access x without an instance of MyClass.

C Variable x is out of scope in the main method.

D The println method expects a String, not an int.

Question 5 – Inheritance Error

```
public class Foo {
  protected int x;

  public Foo(int x) {
    this.x = x;
  }
}
```

```
public class Bar extends Foo {

  public Bar(int x) {
    this.x = x;
  }
}
```

These classes generate a compiler error in the Bar constructor. Why?

A Foo does not have a default constructor.

B Bar does not have a default constructor.

C Bar does not have a member named x.

D Bar does not have access to the x variable in Foo.

Question 5 – Inheritance Error

```
public class Foo {
  protected int x;

  public Foo(int x) {
    this.x = x;
  }
}
```

```
public class Bar extends Foo {

  public Bar(int x) {
    this.x = x;
  }
}
```

These classes generate a compiler error in the Bar constructor. Why?

A Foo does not have a default constructor.

B Bar does not have a default constructor.

C Bar does not have a member named x.

D Bar does not have access to the x variable in Foo.