Lab 21 Written Portion

The following questions ask you to write down the output of some given code. All the code involves three classes: `Base`, `Sub`, and `Main`. The contents of these classes depends on the question.

If you're unsure about the output, then write, compile, and run the code. This handout is also digitally available online so you can easily copy-and-paste it. Just be sure to understand why the output is what it is before moving on.

Question 1: `extends` Without Overriding

Consider the following code:

```java
public class Base {
  public void method() {
    System.out.println("Base");
  }
}

public class Sub extends Base {}

public class Main {
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    base.method();
    sub.method();
    base2.method();
  }
}
```

What is the output of the `main` method above?

Question 2: `extends` With Overriding

Consider the following code:

```java
public class Base {
  public void method() {
    System.out.println("Base");
  }
}

public class Sub extends Base {
  public void method() {
    System.out.println("Sub");
  }
}

public class Main {
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    base.method();
    sub.method();
    base2.method();
  }
}
```

What is the output of the `main` method above?

Question 3: Polymorphism Without Overloading

Consider the following code:

```
public class Base {}

public class Sub extends Base {}

public class Main {
  public void method(Base base) {
    System.out.println("Base");
  }
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    method(base);
    method(sub);
    method(base2);
  }
}
```

What is the output of the `main` method above?

Question 4: Polymorphism With Overloading

Consider the following code:

```
public class Base {}

public class Sub extends Base {}

public class Main {
  public void method(Base base) {
    System.out.println("Base");
  }
  public void method(Sub sub) {
    System.out.println("Sub");
  }
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    method(base);
    method(sub);
    method(base2);
  }
}
```

What is the output of the `main` method above?

Question 5: `extends` With Overloading

Consider the following code:

```
public class Base {
  public method(Base base) {
    System.out.println("Base");
  }
  public method(Sub sub) {
    System.out.println("Sub");
  }
}

public class Sub extends Base {}

public class Main {
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    base.method(base);
    base.method(sub);
    base.method(base2);
    sub.method(base);
    sub.method(sub);
    sub.method(base2);
    base2.method(base);
    base2.method(sub);
    base2.method(base2);
  }
}
```

What is the output of the `main` method above?

Question 6: `extends` With Overloading and Overriding

Consider the following code:

```java
public class Base {
  public method(Base base) {
    System.out.println("Base1");
  }
  public method(Sub sub) {
    System.out.println("Sub1");
  }
}

public class Sub extends Base {
 public method(Base base) {
    System.out.println("Base2");
  }
  public method(Sub sub) {
    System.out.println("Sub2");
  }
}

public class Main {
  public static void main(String[] args) {
    Base base = new Base();
    Sub sub = new Sub();
    Base base2 = sub;

    base.method(base);
    base.method(sub);
    base.method(base2);
    sub.method(base);
    sub.method(sub);
    sub.method(base2);
    base2.method(base);
    base2.method(sub);
    base2.method(base2);
  }
}
```

What is the output of the `main` method above?