

Class 3

1.

Write a program with the following:

- Create a class named Horse
- Inside Horse, create a private String variable named "name" and a private int variable named "age"
- Create a constructor in class Horse with all the parameters.
- Create another constructor (any!).
- Create getters and setters for all variables.
- Create a Horse object from main using any constructor.
- Print horse name from main class.

2.

What will be the output (printed) of the following program?

```
public class Main {
    public static void main(String[] args) {
        Person person = new Person("josh");
        person.setName("sam");
        person.printName(person.getName());
    }
}

public class Person {
    String name;

    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void printName(String name) {
        System.out.println("Name is: "+name);
    }
}
```

3.

In the example below **this** parameter is missing from setName() method.

Will the output be different from question 2 (above)? If so, why?

```
public class Person {
    String name;

    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        name = name;
    }

    public void printName(String name) {
        System.out.println("Name is: "+name);
    }
}
```

4.

Write a program with the following:

- Create a class “Pony” that extends class Horse.
- Add to “Pony” class another method called **bite()** which prints “yamm”.

5.

- Create a class WhitePony that extends Pony class.
- Create a method that return the Pony color (White).
- In your main program, create an object of WhitePony class, call **bite()** method and print the pony color.

6.

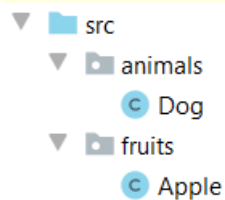
What will the reason for a compiler error with the below code?

```
public class Inheritance {  
  
    private class X {  
        //Class X Members  
    }  
  
    private class Y {  
        //Class Y Members  
    }  
  
    private class Z extends X, Y {  
        //Class Z Members  
    }  
}
```

7.

Given the below project structure:

```
package fruits;
```



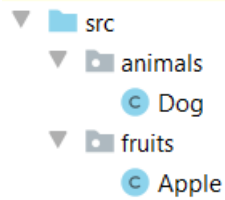
- Why the below code will not compile (run)?

```
public class Apple {  
    public void callDog(){  
        Dog d = new Dog();  
    }  
}
```

- How can it be fixed?

8.

Given the below project structure:



- Why the below code will not compile (run)?

```
package fruits;

import animals.Dog;

public class Apple {
    public void callDog(){
        Dog d = new Dog();
        d.bark();
    }
}
```

```
package animals;

public class Dog {
    protected void bark(){
        System.out.println();
    }
}
```

- How can it be fixed?

9.

What will be the below program output?

```
1 package animals;
2
3 public class Dog {
4     public void bark(){
5         System.out.println("Whaf - Dog");
6     }
7 }
8

1 package animals;
2
3 public class Puppy extends Dog {
4     public void bark(){
5         System.out.println("Whaf - Puppy");
6     }
7 }
8

1 import animals.Dog;
2 import animals.Puppy;
3
4 public class Main {
5     public static void main(String[] args) {
6         Dog rexi = new Puppy();
7         rexi.bark();
8     }
9 }
```

10.

What is the importance of encapsulation in Java?

Challenges:

11.

- What is “Cyclic inheritance”?
- Is it allowed in Java?
- Write one (ignore any errors).

12. - <https://docs.oracle.com/javase/tutorial/java/concepts/interface.html>

Read about java Interfaces and use one in your code.