CS164 – Polymorphism Worksheet 2

Name(s):____

Analyze the classes below and understand what they do in order to implement what is requested.

```
public class Animal {
    private String name;
    private int yearsOld;
    public Animal(){
        this("unknow", 0);
    }
    public Animal(String name, int yearsOld){
        this.name = name;
        this.yearsOld = yearsOld;
    }
    public String getName(){
        return name;
    }
    public int getYearsOld(){
        return yearsOld;
    }
    public void eat(){
         System.out.println("Animals Eat");
    }
    public String toString(){
        return String.format("Name %s, Years Old: %d", name, yearsOld);
    }
}
```

```
public class Carnivores extends Animal {
    public Carnivores(){
        super();
    }
    public Carnivores(String name, int years){
        super(name, years);
    }
    public void eat(){
        System.out.println("Carnivores Eat meat");
    }
}
```

```
public class Herbivores extends Animal {
    public Herbivores(){
        super();
    }
    public Herbivores(String name, int years){
        super(name, years);
    }
    public void eat(){
        System.out.println("Herbivores Eat Plants");
    }
}
```

```
public class Omnivores extends Animal {
    public Omnivores(){
        super();
    }
    public Omnivores(String name, int years){
        super(name, years);
    }
    public void eat(){
        System.out.println("Omnivores Eat Plants and meat");
    }
}
```

```
import java.util.ArrayList;
import java.util.Scanner;
public class AppAnimal {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Animal> list = new ArrayList<>();
        int op = menu(sc);
        while(op!= 5){
            System.out.println("Enter the name of the animal: ");
            sc.nextLine();//needed to not skip reading the name
            String name = sc.nextLine();
            System.out.println("Enter the animal years old: ");
            int years = sc.nextInt();
            Animal a = null;
            switch(op){
                case 1: a = new Animal(name, years);
                    break;
                case 2: a = new Herbivores(name, years);
                    break;
                case 3: a = new Omnivores(name, years);
                    break;
                case 4: a = new Carnivores(name, years);
            }
            list.add(a);
            op = menu(sc);
        }
        print(list);
        ArrayList<String> animals = animalsNames(list);
        System.out.println(animals);

    }
    public static int menu(Scanner sc){
        int op = 0;
        do{
            System.out.println("Menu");
            System.out.println("1 - Enter general Animal");
            System.out.println("2 - Enter Animal Herbivores");
            System.out.println("3 - Enter Animal Omnivores");
            System.out.println("4 - Enter Animal Carnivores");
            System.out.println("5 - Exit");
            System.out.println("Enter your option:");
            op = sc.nextInt();
        }while(op < 1 || op > 5);
        return op;
    }
    public static void print(ArrayList<Animal> lst){
        for (Animal elem : lst) {
            System.out.println(elem);
            elem.eat();
        }
    }
    public static ArrayList<String> animalsNames(ArrayList<Animal> lst){
        ArrayList<String> ret = new ArrayList<>();
        for(Animal animal: lst){
            ret.add(animal.getName());
        }
        return ret;
    }
}
```

1. Implement a method that returns an ArrayList containing only animals that are Herbivores. Call this method inside your main method.
2. Implement a method that return all animals that are older than the average age from the total of animals. Call this method inside your main method.
3. Implement a method that counts and prints the total number of herbivores, omnivores, and carnivores. Call this method inside your main method.
4. Implement a method that returns the ages of all omnivores. Call this method inside your main method.