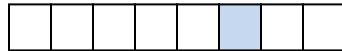


Assignment 100₂/4₈/4₁₆

Using toString and class variables and exceptions

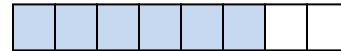
Binary



Ones Comp



Twos Comp



Employee Object



Behaviors

```
get_SS#()
get_Gender()
get_Date_of_Birth()
```

Message - get_SS#()

Payroll Object



1. The toString method is inherited from Object, which all java classes are descendents from. Redefine toString for the classes you developed to display the information from each object. Test the methods using your test classes.
2. Class variables are used to keep data that is shared among all instances of a class. MUST BE PUBLIC! Add class variables to each of your classes. Keep track of the total number of employees ever created AND the total salary for all employees. Each time a constructor is called, update the class variables. Have a method to print out the totals to the screen. Do the same for the car and student classes. Test

from the test classes.

But how do I get access to a class variable?

```
public class EmployeeTest {  
    public static void main(String[] args) {  
        System.out.println("Before adding employees");  
        System.out.println("Total Emp" + Employee.totalEmployees);  
        System.out.println("Total Salaries" + Employee.totalSalary);  
        Employee joe = new Employee("Ian Fisher",52000);  
        Employee joe = new Employee("Bobby Fisher",102000);  
        System.out.println("After adding employees");  
        System.out.println("Total Emp" + Employee.totalEmployees);  
        System.out.println("Total Salaries" + Employee.totalSalary);  
    }  
}
```

NOTE: although these are not covered on the ap exam, you can write a method for a class that will get called right before the object is destroyed or garbage collected.

```
public void finalize()  
{  
    //reduce object count here  
}
```

Optional: add the finalize method to your classes to adjust your total salary and employee count

3. Add exception handling to your classes. Throw illegal argument exceptions if the parameters are illegal. Pick one of your mutator methods for each class and modify it so that it throws an `IllegalArgumentException`. Modify your test classes to use a try catch block to handle the exception.

Below is an example;

//From Employee.java file

```
public void setSalary(double sal) throws IllegalArgumentException  
{  
    if (sal>=0)  
    {  
        salary = sal;  
    }  
    else throw new IllegalArgumentException("Can't have negative salaries");  
}
```

```

    }
    //From EmpTest.java file
    public class EmpTest
    {
        public EmpTest()
        {
            Employee e1 = new Employee();
            System.out.println(e1.getEmployees());
            try
            {
                e1.setSalary(-10);
            } catch (IllegalArgumentException ia)
            {
                System.out.println(ia);
            }
        }
    }

```

Project Name	Assign 4 – Enhancing Emp Car Student
Class 1 Name	Employee
Class 2 Name	EmployeeTest
Class 3 Name	Car
Class 4 Name	CarTest
Class 5 Name	Student
Class 6 Name	StudentTest

Rubric	
Redefine toString in 3 classes	15
Test toString	5
Define class variables in 3 classes (2 for employee 1 for salary, 1 for num employees)	20
Test class vars	20
Modify 3 setters or constructors to throw exceptions	15
Test exceptions	10
Comments	15
TOTAL	100

Recursion*Linear Search*Binary Search*Marine Biology Case Study*Infix*Postfix*Prefix*nlogn