Guided Lab -303.10.1 - Inheritance and Object Type Casting

Lab Overview:

In this lab, we will demonstrate more in-depth how to use Object type-casting and inheritance using Java.

Objective:

By the end of this lab, learners will be able to:

- Describe inheritance in Java
- Identify the object Type casting in Java
- Utilize Inheritance and Object Type Casting

Instructions:

Consider the following classes:

Create a class named **Person,** and write the code below.

```
public class Person {
   String         name;
   static int     lifeSpan = 60;
   static double  ageFactor = 1.0;

   public Person() {

       name = "";
   }
   public Person(String aName) {
       name = aName;
   }
   public String getName() { return name; }
   public void setName(String aName) { name = aName; }
   public String toString() {
       return("Hello, my name is " + name);
   }
   public String talk() {
       return("I have nothing to say.");
```

```
    }
    public String walk() {
        return("I have nowhere to go.");
    }
    public static double lifeSpan() {
        return(lifeSpan * ageFactor);
    }
}
```

Create a class named **Boy,** and write the code below.

```
public class Boy extends Person {
    static double   ageFactor = 1.1;

    public String talk() {
        return(super.talk() + " ... but I love Java class.");
    }
    public String walk() {
        return("I am now walking");
    }
}
```

Create a class named **Girl,** and write the code below.

```
public class Girl extends Person {
    static double   ageFactor = 1.3;
    public Girl(String aName) {
        name = "Ms." + aName;
    }

    public String talk() {
        return("Hello! " + jump());
    }
    public String jump() {
        return("I am jumping.");
    }
    public static double lifeSpan() {
        return(lifeSpan * ageFactor);
    }
}
```

Create a class named **TestPeople,** and write the code below

```
public class TestPeople {
   public static void main(String args[]) {
        Person aPerson;
        Boy jimmy;
        Girl betty;
        aPerson = new Person("Fred");
        jimmy = new Boy();
        betty = new Girl("Betty");

        // Boy b = new Person();  // Throw Error
        // Girl g = new Person()  // Throw Error
        System.out.println(aPerson);
        System.out.println(aPerson.talk());
        System.out.println(aPerson.walk());
        System.out.println();

        System.out.println(jimmy);
        System.out.println(jimmy.talk());
        System.out.println(jimmy.walk());
        System.out.println();

        System.out.println(betty);
        System.out.println(betty.talk());
        System.out.println(betty.walk());
        System.out.println();

        System.out.println((Person)jimmy);
        System.out.println(((Person)jimmy).talk());
        System.out.println(((Person)jimmy).walk());
        System.out.println();

        System.out.println((Person)betty);
        System.out.println(((Person)betty).talk());
        System.out.println(((Person)betty).walk());
        System.out.println();

        System.out.println(Person.lifeSpan());
        System.out.println(Boy.lifeSpan());
        System.out.println(Girl.lifeSpan());

        System.out.println(((Boy)aPerson).talk());
   }
}
```

Output:

```
Hello, my name is Fred.
I have nothing to say.
I have nowhere to go.

Hello, my name is
I have nothing to say... but I love Java class.
I am now walking.

Hello, my name is Ms.Betty.
Hello! I am jumping.
I have nowhere to go.

Hello, my name is
I have nothing to say... but I love Java class.
I am now walking.

Hello, my name is Ms.Betty
Hello! I am jumping.
I have nowhere to go.

60.0
60.0
78.0

Exception in thread "main" java.lang.ClassCastException:
objectTypeCasting.Person cannot be cast to objectTypeCasting.Boy
        at objectTypeCasting.TestPeople.main(TestPeople.java:44)
```

The **lifespan()** method did not work in the way expected. That is because for class methods, method look-ups occur at compile time. The **lifeSpan()** method in the **Person** class is used by both the **Boy** and **Person** classes. In this case, since the method is static and is declared in the **Person** class, the **ageFactor** from the

Person class is used. However, the **Girl** class has its own **lifeSpan()** method, so the **ageFactor** within the **Girl** class is used in that case.

Submission Instructions:

Include the following deliverables in your submission -

- Submit your source code using the Start Assignment button in the top right corner of the assignment page in Canvas.