CST8132 Object Oriented Programming

Exercise 5 – Overloaded constructors, composition, StringBuilder

Due in the lab period of week 9

Steps:

- In this lab exercise you will practice using overloaded constructors, composition, and StringBuilder
- Create a class Name that has three String fields: first middle last
- Put in overloaded constructors that chain calls to the one accepting three String references
  - Name(String, String, String) // first, middle, last
  - Name(String, String) // first, last (set middle to "unknown")
  - Name(String) // first (set middle and last to "unknown")
  - Name() // set all three fields to "unknown"
- Add getter methods for each of the fields (i.e. create Read-Only Properties) (no setters)
- Add a method getFullName()
  - Use a StringBuilder to append the three parts of the name together, separated by spaces
  - Return a reference to a String generated from the StringBuilder
- Add a class person with a field of type Name "A person has-a name"
- There should only be two fields in class Person, one of type Name, another for age of type int
- Put overloaded constructors into class Person
  - One that allows first, middle, and last names to be passed as Strings
  - One that accepts a Name object reference, and an age as an int
    - Make a new Name inside Person, copying the references for the parts of the name.
- Add one method named details() to return a String consisting of the persons full name and age also using a StringBuilder
- Test your classes Person and Name using the following class:

```
public class PersonTester {
    public static void main(String[] args) {
        Person person1 = new Person("a1", "b1", "c1", 11);
        Person person2 = new Person(new Name("a2", "b2", "c2"), 22);
        Person person3 = new Person(new Name("a3", "c3"), 33);
        Person person4 = new Person(new Name("a4"), 44);
        Person person5 = new Person(new Name(), 55);
        System.out.println(person1.details());
        System.out.println(person2.details());
        System.out.println(person3.details());
        System.out.println(person4.details());
        System.out.println(person5.details());
    }
}
```

Grading Guide (Total Score 10):

Name class correct	4
Person class correct	4
Program produces correct output	2

Sample Program Output

```
a1 b1 c1 age: 11
a2 b2 c2 age: 22
a3 unknown c3 age: 33
a4 unknown unknown age: 44
unknown unknown unknown age: 55
```