CSCI 1101 – Winter 2017
Laboratory No. 4

SOLUTIONS (Note: Only solution outlines for relevant exercises are given here).

Exercise 1:

```
//RoomDimension.java
public class RoomDimension
{
      private double length;
      private double width;

      public RoomDimension(double l, double w)
      {
            length = l;
            width = w;
      }
      public double getLength()
      {
            return length;
      }
      public double getWidth()
      {
            return width;
      }
      public double getArea()
      {
            return length*width;
      }

      public String toString()
      {
            String str;
            str = "Length: " + length + "\tWidth: " + width;
            return str;
      }
}

//RoomCarpet.java
public class RoomCarpet
{
      private RoomDimension size;
      private double carpetCost;

      public RoomCarpet(RoomDimension dim, double cost)
      {
            size = new RoomDimension(dim.getLength(), dim.getWidth());
            carpetCost = cost;
      }
      public double getTotalCost()
      {
            return (size.getArea()*carpetCost);
      }
      public String toString()
      {
            String str;
            str = "" + size + "\tTotal Cost: " + getTotalCost();
            return str;
      }
}

//RoomCarpetDemo.java
public class RoomCarpetDemo
{
      public static void main(String[] args)
      {
            RoomDimension room1 = new RoomDimension(10.0,20.0);
            RoomCarpet room1carpet = new RoomCarpet(room1, 25.00);
            System.out.println(room1carpet);
      }
}
```

Exercise 2.

```
//Coin.java
public class Coin
{
      private String name;
      private int value;

      public Coin(String n, int v)
      {
            name = n;
            value = v;
      }
      public String getName()
      {
            return name;
      }
      public int getValue()
      {
            return value;
      }

      public String toString()
      {
            String str;
            str = "Coin: " + name + "\tValue: " + value;
            return str;
      }
}

//Wallet.java
public class Wallet
{
      private Coin[] coin_array;
      private int capacity;
      private int num;
      private int totalValue;
```

```
	public Wallet(int cap)
	{
		coin_array = new Coin[capacity];
		capacity=cap;
		num=0;
		totalValue=0;
	}
	public void addCoin(Coin c)
	{
		if (num==capacity)
		{
			System.out.println(“Cannot add any more coins”);
		}
		else
		{
			coin_array[num] = c;
			num++;
			totalValue = totalValue + c.getValue();
		}
	}
	public Coin removeCoin()
	{
		Coin c;
		if (num==0)
			System.out.println(“Cannot remove. No coins in Wallet”);
		else
		{
			c = coin_array[num-1];
			num--;
			totalValue = totalValue - c.getValue();
		}
		return c;
	}
	public String toString()
	{
		return ("Total value in wallet: " + totalValue);
	}
}

//WalletDemo.java
public class WalletDemo
{
	public static void main(String[] args)
	{
		Coin dollar = new Coin("dollar", 100);
		Coin quarter = new Coin("quarter", 25);
		Wallet myWallet = new Wallet(10);
		myWallet.addCoin(dollar);
		myWallet.addCoin(quarter);
		myWallet.addCoin(quarter);
		System.out.println(myWallet.removeCoin());
		System.out.println(myWallet);
	}
}
```

Exercise 3:

```
public class Person
{
      private String name;

      public Person()
      {
      }
      public void setName(String newName)
      {
            name = newName;
      }
      public String getName()
      {
            return name;
      }
      public String toString()
      {
            return ("Name: " + name);
      }
      public boolean hasSameName(Person anotherPerson)
      {
            if (this.name.equals(anotherPerson.getName()))
                  return true;
            else
                  return false;
      }
}

public class Student extends Person
{
      private int studentNumber;

      public Student()
      {
      }
      public void reset(String newName, int newNumber)
      {
            setName(newName);
            studentNumber = newNumber;
      }
      public int getStudentNumber()
      {
            return studentNumber;
      }
      public void setStudentNumber(int n)
      {
            studentNumber = n;
      }
      public String toString()
      {
            return ("Name: " + getName() + "\nStudent Number: " +
studentNumber);
      }
      public boolean equals(Student anotherStudent)
      {
            if ((this.studentNumber == anotherStudent.getStudentNumber())
                  &&(this.hasSameName(anotherStudent)))
                  return true;
            else
                  return false;
      }
}

public class StudentDemo
{
      public static void main(String[] args)
      {
            Student joe1 = new Student();
            joe1.reset("Joe", 123);
            Student joe2 = new Student();
            joe2.reset("Joe", 345);
            System.out.println(joe1);
            System.out.println(joe2);
            if (joe1.hasSameName(joe2))
                  System.out.println("Same name");
            if (joe1.equals(joe2))
                  System.out.println("Wow!");
            else
                  System.out.println("Different IDs");
      }
}
```