Class Donation

```
import java.io.Serializable;
public class Donation implements Serializable{
	private double amount;
	private String location;
	private String donorName;
	public Donation(double amount, String location, String donorName) {
		this.amount = amount;
		this.location = location;
		this.donorName = donorName;
	}
	public Donation(Donation d){
		this.amount = d.amount;
		this.location = d.location;
		this.donorName = d.donorName;
	}

	public double getAmount() throws Exception{
		if(amount < 0)
			throw new Exception("Amount is negative!");
		return amount;
	}
	public String getDonorName() {
		return donorName;
	}
	public void display(){
		System.out.println("Amount: " + amount);
		System.out.println("Location: " + location);
		System.out.println("Donor name: " + donorName);
	}
}
```

Class Cash

```
public class Cash extends Donation{
	private String currency;
	public Cash(double amount, String location, String donorName,
			String currency) {
		super(amount, location, donorName);
		this.currency = currency;
	}
	public Cash(Cash csh){  super(csh); this.currency = csh.currency; }
	public String getCurrency() { return currency;  }
	public void display(){
		super.display();
		System.out.println("Currency: " + currency);
	}
}
```

Class Check

```
public class Check extends Donation{
	private String bankName;
	public Check(double amount, String location, String donorName,
			String bankName) {
		super(amount, location, donorName);
		this.bankName = bankName;
	}
	public Check(Check chk){
		super(chk);
		this.bankName = chk.bankName;
	}
	public String getBankName() {
		return bankName;
	}
	public void display(){
		super.display();
		System.out.println("Bank name: " + bankName);
	}
}
```

Interface IOInterface

```
import java.io.IOException;

public interface IOInterface {

public void saveToFile(String filename, String donor) throws IOException;

public void loadFromFile(String filename,Check[]arrCheck)throws IOException;

}
```

Class CharityAssociation

```
import java.io.*;
public class CharityAssociation implements IOInterface{
	private String name;
	private Donation [] arrDon;
	private int nbDon;
	public CharityAssociation(String name, int size){
		this.name = name;
		arrDon = new Donation[size];
		nbDon = 0;
	}
	public boolean addDonation(Donation d){
		if(nbDon >= arrDon.length)
			return false;
		if(d instanceof Cash)
			arrDon[nbDon++] = new Cash((Cash) d);
		else
			arrDon[nbDon++] = new Check((Check) d);
		return true;
	}
	public double avgCashDonations(String cur){
		double sum = 0;
		int nb = 0;
		for(int i = 0; i < nbDon; i++){
if(arrDon[i] instanceof Cash && ((Cash)arrDon[i]).getCurrency().equals(cur)){
				try{
					sum += arrDon[i].getAmount();
					nb++;
				} catch(Exception e){
					System.out.println(e);
				}
			}
		}
		if(nb != 0)
			return sum / nb;
		return 0;
	}

	public Check getCheck(String bName){
		for(int i = 0; i < nbDon; i++)
			if(arrDon[i] instanceof Check &&
					((Check)arrDon[i]).getBankName().equals(bName))
				return (Check) arrDon[i];
		return null;
	}
```

```
public void saveToFile(String filename, String donor) throws IOException{
            File f = new File(filename);
            FileOutputStream outStream = new FileOutputStream(f);
            ObjectOutputStream outCash = new ObjectOutputStream(outStream);

            for(int i = 0; i < nbDon; i++){
                  if(arrDon[i] instanceof Cash &&
arrDon[i].getDonorName().equals(donor))
                        outCash.writeObject(arrDon[i]);
            }
            outCash.close();
            outStream.close();
      }

      //extra method for testing
      public void saveAll(String filename) throws IOException{
            File f = new File(filename);
            FileOutputStream outStream = new FileOutputStream(f);
            ObjectOutputStream outDon = new ObjectOutputStream(outStream);

            for(int i = 0; i < nbDon; i++)
                  outDon.writeObject(arrDon[i]);

            outDon.close();
            outStream.close();
      }

public void loadFromFile(String filename, Check[] arrCheck) throws
IOException{
            int counter = 0;
            File f = new File(filename);
            FileInputStream inStream = new FileInputStream(f);
            ObjectInputStream inDon = new ObjectInputStream(inStream);
            try{
                  while(true){
                        try{
                              Donation d = (Donation) inDon.readObject();
                              if(d instanceof Check)
                                    arrCheck[counter++] = (Check) d;
                        } catch(ClassNotFoundException e){
                              System.out.println(e);
                        }
                  }
            }catch(EOFException e){
                  System.out.println("Finished reading");
                  inDon.close();
                  inStream.close();
            }
      }
}
```

Class Test

```java
import java.io.*;
public class test {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		CharityAssociation CA = new CharityAssociation("KSU", 3);
		CA.addDonation(new Cash(100, "Riyadh", "Ahmad", "Riyal"));
		CA.addDonation(new Cash(150, "Jeddah", "Ali", "Dollar"));
		CA.addDonation(new Check(80, "Riyadh", "Khalid", "Rajhi"));
		try {
			CA.saveToFile("Cash.data", "Ahmad");
		} catch (IOException e) {
			System.out.println(e);
		}
		try {
			CA.saveAll("donations.data");
		} catch (IOException e) {
			System.out.println(e);
		}
		Check [] chks = new Check[1];
		try {
			CA.loadFromFile("donations.data", chks);
		} catch (IOException e) {
			System.out.println(e);
		}
		chks[0].display();
	}
}
```