1.School Management

```
import java.util.*;
class Student {
	private String name;
	private float percentage;

	public Student() {}

	public Student(String name, float percentage) {
		this.name = name;
		this.percentage = percentage;
	}

	public void setName(String name) {
		this.name = name;
	}

	public void setPercentage(float percentage) {
		this.percentage = percentage;
	}

	public String getName() {
		return name;
	}

	public float getPercentage() {
		return percentage;
	}

	public boolean equals(Object obj) {
		if(obj instanceof Student) {
```

```
                Student s = (Student)obj;
                return s.name.equals(name) && s.percentage == percentage;
            }
            return false;
        }

        public String toString() {
            return name+" "+percentage;
        }

}

class Sorting implements Comparator<Student> {
        public int compare(Student o1, Student o2) {
            return o1.getName().compareTo(o2.getName());
        }
}

class School {
        ArrayList<Student> studentList = new ArrayList<>();
        //      ArrayList<Student> slist = new ArrayList<>();

        public ArrayList<Student> sortByName() {
            String name="";
            //for(Student s : studentList)
            //      if(name<s.getName())
            //              slist.add(s.getName());
            Collections.sort(studentList,new Sorting());
            return studentList;
        }

        public double getAvgPercentage() {
```

```
                double totalPercentage = 0.0;
                double avgPercentage = 0.0;
                int count = 0;
                for(Student s : studentList) {
                        count++;
                        totalPercentage = totalPercentage+s.getPercentage();
                }
                avgPercentage = totalPercentage/count;
                return avgPercentage;
        }
}

class Main {
        public static void main(String[] args) {
                ArrayList<Student> list = new ArrayList<>();
                list.add(new Student("Steve",(float)56.3));
                list.add(new Student("Bob",(float)67.3));
                list.add(new Student("Alice",(float)98.4));
                list.add(new Student("Mark",(float)40));
                School obj = new School();
                obj.studentList = list;
                System.out.println(obj.sortByName());
                System.out.println(obj.getAvgPercentage());
        }
}
```

2.Travel

```
import java.util.*;
import java.util.Map.Entry;
class Travel {
        HashMap<String,Integer> holidayPkg = new HashMap<>();
```

```
	int total=0;
	int count=0;

	public int cheapestPackage(int numberOfPlaces) {
		ArrayList<Integer> alist = new ArrayList<>();
		for(Entry<String,Integer> e : holidayPkg.entrySet())
			alist.add(e.getValue());
		Collections.sort(alist);
		for(int i=0;i<numberOfPlaces;i++)
			total=total+alist.get(i);
		return total;
	}

	public int maximumPlace(int budget) {
		ArrayList<Integer> alist = new ArrayList<>();
		int i=0;
		total=0;
		for(Entry<String,Integer> e : holidayPkg.entrySet())
			alist.add(e.getValue());
		Collections.sort(alist);
		while(total<=budget){
			total=total+alist.get(i);
			count++;
			i++;
		}
		if(count > 0)
			count--;
		return count;
	}
}
```

```
class Main {
        public static void main(String[] args) {
                Travel obj = new Travel();
                obj.holidayPkg.put("Delhi",5000);
                obj.holidayPkg.put("Jaipur",4000);
                obj.holidayPkg.put("Agra",2500);
                obj.holidayPkg.put("Goa",7000);
                System.out.println(obj.cheapestPackage(3));
                System.out.println(obj.maximumPlace(15000));
        }
}
```

3.Adding Tv shows

```
import java.util.*;

class Source {
    public String printIndex(ArrayList<String> list,int ind) {
        return list.get(ind);
    }

    public ArrayList<String> addAfter(ArrayList<String> a, String m, String n)
{
        if(a.contains(m))
            a.add(a.indexOf(m)+1,n);
        return a;
    }
}

class ArrayList1 {
    public static void main(String[] args) {
        ArrayList<String> tvShows = new ArrayList<>();
        tvShows.add("Breaking Bad");
        tvShows.add("GOT");
        tvShows.add("FRIENDS");
        tvShows.add("Prison Break");
        Source s = new Source();
        System.out.println(s.printIndex(tvShows,3));
        System.out.println(s.addAfter(tvShows,"FRIENDS","Supernaturals"));
    }
}
```

4.DrinkingAge (P.S: Not sure)

```
class Age {
	String drinkingAge;
}

class Implementation {
	public String validateIntAgeToDrink (Age obj, int personAge) throws IllegalAgeException {
		try{
			if(personAge < 21)
				throw new IllegalAgeException("Illegal drinking age");
			else
				obj.drinkingAge = "legal";
		}catch(IllegalAgeException e) {
			obj.drinkingAge = "illegal";
			return e.getMessage();
		}
		return obj.drinkingAge;
	}

	public String validateStringAgeToDrink(Age obj,String personAge) throws IllegalAgeException
{
		try {
			int age = Integer.parseInt(personAge);
			if(personAge.length()>2 || age < 21)
				throw new IllegalAgeException("Invalid age detail or drinking age");
			else
				obj.drinkingAge = "legal";
		}
		catch(IllegalAgeException e) {
			obj.drinkingAge = "illegal";
			return e.getMessage();
		}
		return obj.drinkingAge;
```

```
	}
}

class IllegalAgeException extends Exception {
	private String message;
	public IllegalAgeException ( String message) {
		super();
		this.message = message;
	}

	public String getMessage() {
		return message;
	}

	public String toString() {
		return message;
	}
}

class Main {
	public static void main(String[] args) throws IllegalAgeException {
		Age age = new Age();
		Implementation imp = new Implementation();
		System.out.println(imp.validateIntAgeToDrink(age, 25));
		System.out.println(imp.validateStringAgeToDrink(age,"311"));

	}
}
```

5.Country Code

```
class PhoneRepository {
```

```
	public static String getCountryName (String countryCode) throws InvalidCodeException {
		//String message = "";
		//int code = Integer.parseInt(countryCode);
		try{
		if(countryCode.equals("90") || countryCode.equals("91") ||
countryCode.equals("92") || countryCode.equals("93") || countryCode.equals("94") ||
countryCode.equals("95") || countryCode.equals("96") || countryCode.equals("97") ||
countryCode.equals("98") || countryCode.equals("99") || countryCode.equals("100"))
			return "India";
		else if(countryCode.equals("901"))
			return "US";
		else
			throw new InvalidCodeException("No country with given code found");
		}catch(InvalidCodeException e) {
			return e.getMessage();
		}/*catch(Exception e) {
			return e.getMessage();
		}*/
		//return message;
	}

}

class Client {
	public static String getCountry(String countryCode) throws InvalidCodeException {
		try{
			if(countryCode.length() > 3 || countryCode.length() < 2)
				throw new InvalidCodeException("Invalid code detail found");
			else {
				try{
					return PhoneRepository.getCountryName(countryCode);
				}catch(InvalidCodeException e) {
					return e.getMessage();
				}
```

```
                }
            }catch(InvalidCodeException e) {
                return e.getMessage();
            }
        }
}

class InvalidCodeException extends Exception {
        private String msg;

        public InvalidCodeException(String msg) {
            super();
            this.msg = msg;
        }

        public String getMessage() {
            return msg;
        }

        public String toString() {
            return msg;
        }


}

class Main {
        public static void main(String[] args) throws InvalidCodeException {
            //try{
                System.out.println(PhoneRepository.getCountryName(""));
                System.out.println(Client.getCountry("1020"));
            /*}catch(InvalidCodeException e) {
```

```
                    System.out.println(e.getMessage());
                }*/
        }
}
```