**GENERICS**

1. Use a HashSet to hold Employee Objects. Upon running the application, the details of the employees added to the HashSet should be displayed.

Employee <<class>>

|-- id

|-- name

|-- salary

|-- department

|-- displayDetails()

Feel free to add properties and methods to Employee Class

**Note:** if we try to store any object other than Employee Object in HashSet, we should not be allowed to.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Write an application to hold 10 random int values as keys and 10 random double values as values for a HashMap. Print the data store in the HashMap. Note: Keys can only be int and values double.

Graphical user interface, text, application

Description automatically generated

1. Write a generic method to exchange the positions of two different elements in an array.

Graphical user interface, text, application

Description automatically generated

1. Design a class named Pair which has two properties. The name of the first property is key and that of the second property is value. When designing the class take case of the following scenarios:
2. Create an Object of Pair class to store String value for the property key and String value for the property value. Restriction Apart from String type no other types should be acceptable as Key or value input

e.g. myObj.setKey(“1”);

myObj.setValue(“Hello”);

b. Create an object of the class Pair to store String value for the property key and java.util.Date as value for the property value

myObj.setKey(“Today is”);

myObj.setValue(new java.util.Date());

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated