Lab program 8

```
package practice;
class WrongAge extends RuntimeException
{
	WrongAge(String msg){
		super(msg);
	}
}
class Father
{
	int age;
	Father(int age1)
	{
		age=age1;
		System.out.println("Father age:"+age);
	}
}
class Son extends Father
{
	Son(int age1)
	{
		super(age1);
		System.out.println("Son age:"+age);
	}
}
public class Age{
	public static void main(String args[]) throws WrongAge
	{
	//it is not good to use throw in the main of the calling because it will pass it to jvm and program will terminate abruptly
	//throws will tell the calling function abt the exception that can be produced by this object
```

```
		//this removes the need of the using try and catch
		//the function calling main have to handle the exception
		// throws keyword also deals with compile time error
		int i=args.length;
			int j=Integer.parseInt(args[0]);
			int k=Integer.parseInt(args[1]);
			try {
				if(i<=0 || k>j)
				{
					throw new WrongAge("Son age can't be greater than Father"); //this will create the exception object and the jvm will search if the exception is handled or not
				}
				else
				{
					Father f=new Father(j);
					Son s=new Son(k);
				}
			}
			catch (WrongAge e) {
				// e.getMessage() //print only the message
				// e.toString()
				e.printStackTrace(); //this is done to print the message ,description ,and stacktrace of the error
			}
		}
}

package practice;
class Temp<Type>{
   Type value;

   Temp(){
```

```
    }
    Temp(Type value){
        this.value=value;
    }
    public Type getValue() {
        return value;
    }

    public void setValue(Type value) {
        this.value = value;
    }
}
public class Generics {
    public static void main(String[] args) {
        Temp<Float> test=new Temp<Float>(0.24f);
        System.out.println(test.getValue());
        test.setValue(0.36f);
        System.out.println(test.getValue());
        Temp<String> stest=new Temp<String>("Hello");
        System.out.println(stest.getValue());
    }
}
```