4.1. Do the workshop 2 in the CD (Digital signatures).

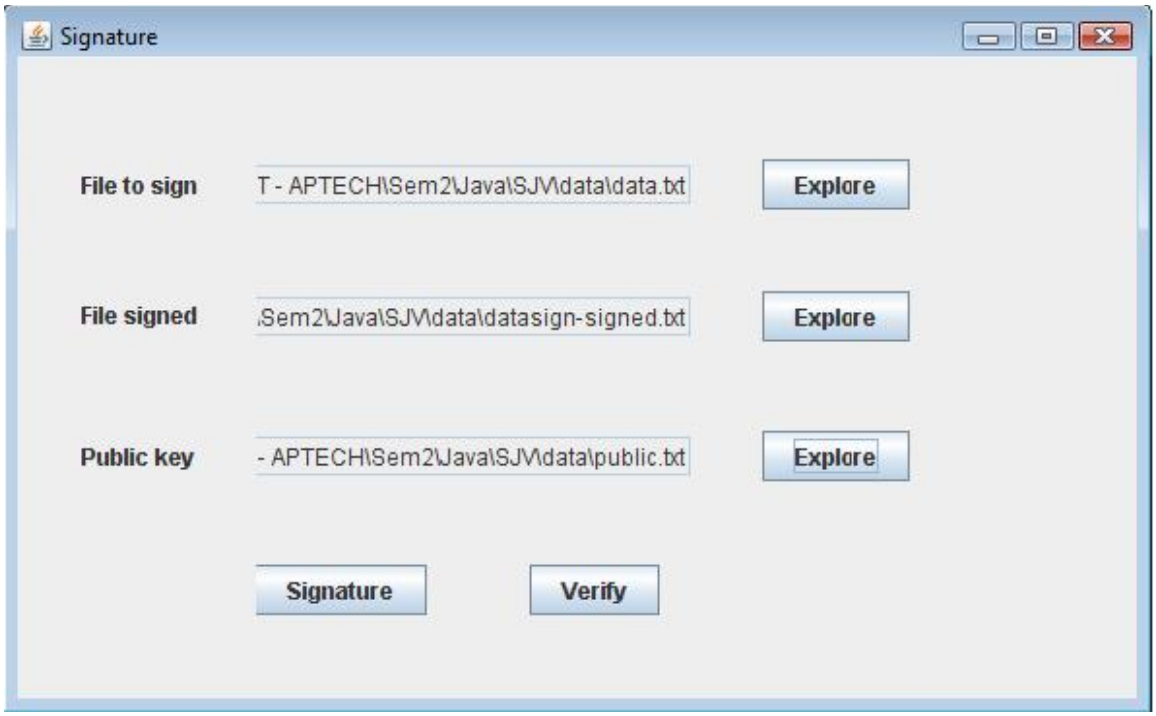
4.2. Type the below code

```
   public void doSignature()
   {
      try
      {
         KeyPairGenerator g = KeyPairGenerator.getInstance("DSA","SUN");
         SecureRandom random = SecureRandom.getInstance("SHA1PRNG","SUN");
         g.initialize(1024,random);
         KeyPair keys = g.genKeyPair();
         PublicKey publickey = keys.getPublic();
         PrivateKey privatekey = keys.getPrivate();
         java.security.Signature asign =
java.security.Signature.getInstance("SHA1withDSA","SUN");
         asign.initSign(privatekey);
         FileInputStream fin = new FileInputStream(fsign);
         byte [] data = new byte[1024];
         while(fin.available()!=0)
         {
            int len = fin.read(data);
            asign.update(data,0,len);
         }
         fin.close();
         FileOutputStream fout = null;
         fout = new FileOutputStream(fpublic);
         fout.write(publickey.getEncoded());
         fout.close();
         fout = new FileOutputStream(fsigned);
         fout.write(asign.sign());
         fout.close();
         JOptionPane.showMessageDialog(null,"Sign to
successful","Signature",JOptionPane.INFORMATION_MESSAGE);
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
   public void doVerify()
   {
      try
      {
         FileInputStream fin = new FileInputStream(fpublic);
         byte [] encodekey = new byte[fin.available()];
         fin.read(encodekey);
```

```
            fin.close();
            X509EncodedKeySpec spec = new X509EncodedKeySpec(encodekey);
            KeyFactory keys = KeyFactory.getInstance("DSA","SUN");
            PublicKey publickey = keys.generatePublic(spec);
            java.security.Signature asign =
java.security.Signature.getInstance("SHA1withDSA","SUN");
            asign.initVerify(publickey);
            fin = new FileInputStream(fsigned);
            byte []signtoverify = new byte[fin.available()];
            fin.read(signtoverify);
            fin.close();
            fin = new FileInputStream(fsign);
            byte [] b = new byte[1024];
            while(fin.available()!=0)
            {
               int len = fin.read(b);
               asign.update(b,0,len);
            }
            fin.close();
            boolean  isOk ;
            isOk = asign.verify(signtoverify);
            if(isOk)
               JOptionPane.showMessageDialog(null,"Verifysign
successful","Signature",JOptionPane.INFORMATION_MESSAGE);
            else
               JOptionPane.showMessageDialog(null,"Verifysign
fail","Signature",JOptionPane.INFORMATION_MESSAGE);
         }
         catch(Exception e)
         {
            System.out.println(e);
         }
      }
```

Adding some functions, GUI and executing the program.

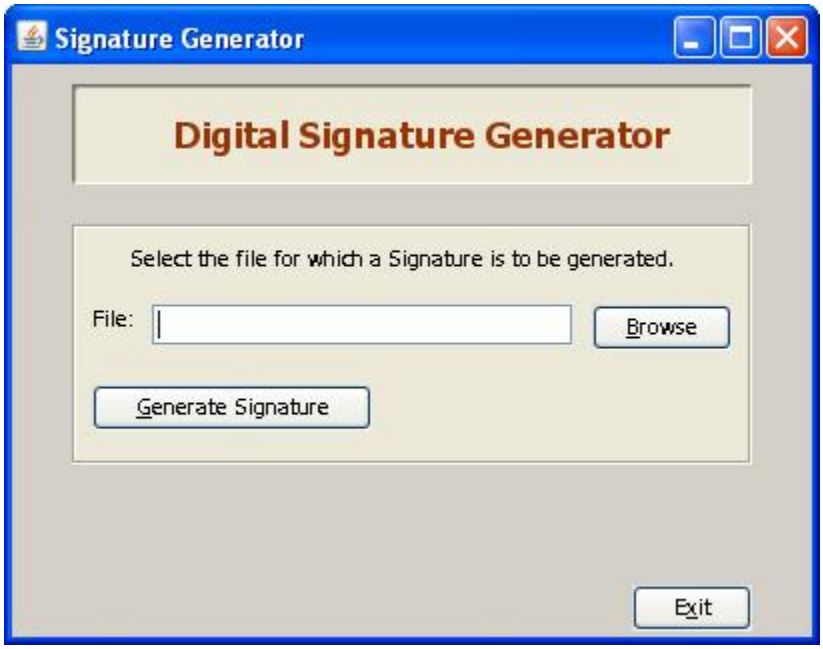
The output of the program is as shown



Do It Yourself

4.3. Write a Java application generate digital signature and public key

The output of the program is as shown as below.



4.4. Write another java program to validate signature which has been generated by exercise 4.1.

The output of the program is as shown as below.