Zip a Folder or a Directory in Java

When we zip a file or directory , it’s size reduces because files are compressed based on zip algorithm.

Following is the code to zip a directory or folder in java.

This code only makes a zip file of a directory and not a single file.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.zip.ZipEntry;
import java.util.zip.ZipOutputStream;

public class Zipfile {

   List<String> filesListInDir = new ArrayList<String>();

   public static void main(String[] args) {

      File dir = new File("c:\\temp");
      String zipDirName = "c:\\temp1\\tempabc.zip";

      Zipfile zipFiles = new Zipfile();
      zipFiles.zipDirectory(dir, zipDirName);
   }

   /**
    * This method zips the directory
    * @param dir
```

```
     * @param zipDirName
     */
    private void zipDirectory(File dir, String zipDirName) {
        try {
            populateFilesList(dir);
            //now zip files one by one
            //create ZipOutputStream to write to the zip file
            FileOutputStream fos = new FileOutputStream(zipDirName);
            ZipOutputStream zos = new ZipOutputStream(fos);
            for(String filePath : filesListInDir){
                System.out.println("Zipping "+filePath);
                //for ZipEntry we need to keep only relative file path, so we used substring on absolute path
                ZipEntry ze = new ZipEntry(filePath.substring(dir.getAbsolutePath().length()+1,
filePath.length()));
                zos.putNextEntry(ze);
                //read the file and write to ZipOutputStream
                FileInputStream fis = new FileInputStream(filePath);
                byte[] buffer = new byte[1024];
                int len;
                while ((len = fis.read(buffer)) > 0) {
                    zos.write(buffer, 0, len);
                }
                zos.closeEntry();
                fis.close();
            }
            zos.close();
            fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```
/**
 * This method populates all the files in a directory to a List
 * @param dir
 * @throws IOException
 */
private void populateFilesList(File dir) throws IOException {
    File[] files = dir.listFiles();
    for(File file : files){
        if(file.isFile()) filesListInDir.add(file.getAbsolutePath());
        else populateFilesList(file);
    }
}

/**
 * This method compresses the single file to zip format
 * @param file
 * @param zipFileName
 */
private static void zipSingleFile(File file, String zipFileName) {
    try {
        //create ZipOutputStream to write to the zip file
        FileOutputStream fos = new FileOutputStream(zipFileName);
        ZipOutputStream zos = new ZipOutputStream(fos);
        //add a new Zip Entry to the ZipOutputStream
        ZipEntry ze = new ZipEntry(file.getName());
        zos.putNextEntry(ze);
        //read the file and write to ZipOutputStream
        FileInputStream fis = new FileInputStream(file);
        byte[] buffer = new byte[1024];
        int len;
```

```
            while ((len = fis.read(buffer)) > 0) {
                zos.write(buffer, 0, len);
            }

            //Close the zip entry to write to zip file
            zos.closeEntry();
            //Close resources
            zos.close();
            fis.close();
            fos.close();
            System.out.println(file.getCanonicalPath()+" is zipped to "+zipFileName);

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

}
```