**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HelloWorld.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package oata;

public class HelloWorld {

public static void main(String[] args) {

System.*out*.println("Hello World Vaishali............");

}

}

**\*\*\*\*\*\*\*In Project Folder\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project default=*"main"* >

<property

name=*"src.dir"*

value=*"src"* />

<property

name=*"build.dir"*

value=*"build"* />

<property

name=*"classes.dir"*

value=*"${build.dir}/classes"* />

<property

name=*"jar.dir"*

value=*"${build.dir}/jar"* />

<property

name=*"main-class"*

value=*"oata.HelloWorld"* />

<target name=*"clean"* >

<delete dir=*"${build.dir}"* />

</target>

<target name=*"compile"* >

<mkdir dir=*"${classes.dir}"* />

<javac

destdir=*"${classes.dir}"*

srcdir=*"${src.dir}"* />

</target>

<target

name=*"jar"*

depends=*"compile"* >

<mkdir dir=*"${jar.dir}"* />

<jar

basedir=*"${classes.dir}"*

destfile=*"${jar.dir}/${ant.project.name}.jar"* >

<manifest>

<attribute

name=*"Main-Class"*

value=*"${main-class}"* />

</manifest>

</jar>

</target>

<target

name=*"run"*

depends=*"jar"* >

<java

fork=*"true"*

jar=*"${jar.dir}/${ant.project.name}.jar"* />

</target>

<target

name=*"clean-build"*

depends=*"clean,jar"* />

<target

name=*"main"*

depends=*"clean,run"* />

</project>

\*Demo-2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**package** com.patani;

**public** **class** HelloAnt

{

**public** **static** **void** main(String[] args)

{

System.*out*.println("Hello Ant..."+"Vaishali");

}

}

\*\*\*\*\*\*\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<?xml version=*"1.0"* ?>

<project default=*"main"*>

<target name=*"main"* depends=*"compile, run"* description=*"Main target"*>

<echo>

Building the .jar file.

</echo>

</target>

<target name=*"compile"* description=*"Compilation target"*>

<javac srcdir=*"./src"* destdir=*"build"*/>

</target>

<target name=*"run"* description=*"Run target"*>

<java classname=*"com.patani.HelloAnt"*>

<classpath path=*"build"*/>

</java>

</target>

</project>

<!-- Create Build folder in Javva project -->

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AntDemo3\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*MyMath.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**package** test;

**import** math.MyMath;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

MyMath math = **new** MyMath();

System.*out*.println("Result is: " + math.multi(5, 10));

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project default=*"main"*>

<!-- Sets variables which can later be used. -->

<!-- The value of a property is accessed via ${} -->

<property name=*"src.dir"* location=*"src"* />

<property name=*"build.dir"* location=*"bin"* />

<property name=*"dist.dir"* location=*"dist"* />

<property name=*"docs.dir"* location=*"docs"* />

<!-- Deletes the existing build, docs and dist directory-->

<target name=*"clean"*>

<delete dir=*"${build.dir}"* />

<delete dir=*"${docs.dir}"* />

<delete dir=*"${dist.dir}"* />

</target>

<!-- Creates the build, docs and dist directory-->

<target name=*"makedir"*>

<mkdir dir=*"${build.dir}"* />

<mkdir dir=*"${docs.dir}"* />

<mkdir dir=*"${dist.dir}"* />

</target>

<!-- Compiles the java code (including the usage of library for JUnit -->

<target name=*"compile"* depends=*"clean, makedir"*>

<javac srcdir=*"${src.dir}"* destdir=*"${build.dir}"*>

</javac>

</target>

<!-- Creates Javadoc -->

<target name=*"docs"* depends=*"compile"*>

<javadoc packagenames=*"src"* sourcepath=*"${src.dir}"* destdir=*"${docs.dir}"*>

<!-- Define which files / directory should get included, we include all -->

<fileset dir=*"${src.dir}"*>

<include name=*"\*\*"* />

</fileset>

</javadoc>

</target>

<!--Creates the deployable jar file -->

<target name=*"jar"* depends=*"compile"*>

<jar destfile=*"${dist.dir}\de.vogella.build.test.ant.jar"* basedir=*"${build.dir}"*>

<manifest>

<attribute name=*"Main-Class"* value=*"test.Main"* />

</manifest>

</jar>

</target>

<target name=*"run"*>

<java jar=*"${dist.dir}\de.vogella.build.test.ant.jar"* fork=*"true"*/>

</target>

<target name=*"main"* depends=*"compile, jar,run, docs"*>

<description>Main target</description>

</target>

</project>

\*\*\*\*\*\*\*\*\*\*\*\*\*Create dist folder in project\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jdbc Build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<project default="run" basedir=".">

<property name="src.dir" location="src"/>

<property name="lib.dir" location="lib"/>

<property name="build.dir" location="build"/>

<path id="classpath.path">

<!--<fileset dir="${src.dir}"/>-->

<fileset dir="${lib.dir}" includes="ojdbc6.jar"/>

</path>

<target name="clean" description="Clean all build targets">

<delete dir="${build.dir}"/>

</target>

<target name="init">

<mkdir dir="${build.dir}"/>

<mkdir dir="${lib.dir}"/>

<!-- Copy the primary program and modules -->

<copy todir="${lib.dir}">

<fileset dir="C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib" includes="\*.jar"/>

</copy>

</target>

<target name="compile" depends="init" description="Compile program">

<javac srcdir="${src.dir}" destdir="${build.dir}">

<classpath refid="classpath.path"/>

</javac>

</target>

<target name="run" depends="compile" description="Run program">

<java classname="TestJdbcResultSetDemo">

<classpath refid="classpath.path"/>

<classpath>

<pathelement location="${build.dir}"/>

</classpath>

</java>

</target>

<!--<target name="run-base" depends="compile" description="Run program">

<java classname="TestJdbcResultSetDemo">

<classpath refid="classpath.path"/>

<classpath>

<pathelement location="${build.dir}"/>

</classpath>

</java>

</target>-->

</project>

**\*\*\*\*\*\*\*\*\*\*\*Command Line\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" ?>

<project default="main">

<target name="main" depends="compile, run" description="Main target">

<echo>

Building the .jar file.

</echo>

</target>

<target name="compile" description="Compilation target">

<javac srcdir="./src" destdir="build"/>

</target>

<target name="run" description="Run target">

<java classname="TestEmpDemo">

<arg value="70"/>

<classpath path="build"/>

</java>

</target>

</project>

<!-- Create Build folder in Javva project -->

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ant Web Project\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" ?>

<project name="TT" default="war" basedir=".">

<property name="jboss" value="C:/WildFly8.1.0" />

<property name="jboss.deploynment" value="C:/WildFly8.1.0/standalone/deployments" />

<property name="base" value="." />

<path id="compile.classpath">

<fileset dir="WebContent/WEB-INF/lib">

<include name="\*.jar" />

</fileset>

<fileset dir="C:/WildFly8.1.0/modules/system/layers/base/javax/servlet/api/main">

<include name="\*.jar" />

</fileset>

</path>

<target name="init">

<mkdir dir="build/classes" />

<mkdir dir="dist" />

</target>

<target name="compile" depends="init">

<javac destdir="build/classes" debug="true" srcdir="src" includeantruntime="false">

<classpath refid="compile.classpath" />

</javac>

</target>

<target name="war" depends="compile">

<war destfile="dist/TT.war" webxml="WebContent/WEB-INF/web.xml" >

<fileset dir="WebContent" />

<lib dir="WebContent/WEB-INF/lib" />

<classes dir="build/classes" />

</war>

</target>

<target name="deploy" depends="war" description="Deploy application">

<copy todir="${jboss.deploynment}" file="${base}/dist/TT.war">

</copy>

</target>

<target name="clean">

<delete dir="dist" />

<delete dir="build" />

</target>

</project>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MyMath.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package math;

public class MyMath {

public int multi(int number1, int number2)

{ return number1 \* number2;

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TestCalculator.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package test;

import math.MyMath;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class TestCalculator

{

@Test

public void testMulti()

{

MyMath math = new MyMath();

assertEquals(150, math.multi(5, 10));

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0"?>

<project name="Ant-Test" default="main" basedir=".">

<!-- Sets variables which can later be used. -->

<!-- The value of a property is accessed via ${} -->

<property name="src.dir" location="src" />

<property name="build.dir" location="bin" />

<!-- Variables used for JUnit testin -->

<property name="test.dir" location="src" />

<property name="test.report.dir" location="testreport" />

<!-- Define the classpath which includes the junit.jar and the classes after compiling-->

<path id="junit.class.path">

<pathelement location="lib/junit-4.4.jar" />

<pathelement location="lib/easymock-2.5.jar" />

<pathelement location="${build.dir}" />

</path>

<!-- Deletes the existing build, docs and dist directory-->

<target name="clean">

<delete dir="${build.dir}" />

<delete dir="${test.report.dir}" />

</target>

<!-- Creates the build, docs and dist directory-->

<target name="makedir">

<mkdir dir="${build.dir}" />

<mkdir dir="${test.report.dir}" />

</target>

<!-- Compiles the java code (including the usage of library for JUnit -->

<target name="compile" depends="clean, makedir">

<javac srcdir="${src.dir}" destdir="${build.dir}">

<classpath refid="junit.class.path" />

</javac>

</target>

<!-- Run the JUnit Tests -->

<!-- Output is XML, could also be plain-->

<target name="junit" depends="compile">

<junit printsummary="on" fork="true" haltonfailure="yes">

<classpath refid="junit.class.path" />

<formatter type="xml" />

<batchtest todir="${test.report.dir}">

<fileset dir="${src.dir}">

<include name="\*\*/\*Test\*.java" />

</fileset>

</batchtest>

</junit>

</target>

<target name="main" depends="compile, junit">

<description>Main target</description>

</target>

</project>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AntJUnitTestProgect-7-for-repetable-Build\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0"?>

<project name="Ant-Test" default="main" basedir=".">

<!-- Sets variables which can later be used. -->

<!-- The value of a property is accessed via ${} -->

<property name="src.dir" location="src" />

<property name="build.dir" location="bin" />

<!-- Variables used for JUnit testin -->

<property name="test.dir" location="src" />

<property name="test.report.dir" location="testreport" />

<!-- Define the classpath which includes the junit.jar and the classes after compiling-->

<path id="junit.class.path">

<pathelement location="lib/junit-4.4.jar" />

<pathelement location="lib/easymock-2.5.jar" />

<pathelement location="${build.dir}" />

</path>

<!-- Deletes the existing build, docs and dist directory-->

<target name="clean">

<delete dir="${build.dir}" />

<delete dir="${test.report.dir}" />

</target>

<!-- Creates the build, docs and dist directory-->

<target name="makedir">

<mkdir dir="${build.dir}" />

<mkdir dir="${test.report.dir}" />

</target>

<!-- Compiles the java code (including the usage of library for JUnit -->

<target name="compile" depends="clean, makedir">

<javac srcdir="${src.dir}" destdir="${build.dir}">

<classpath refid="junit.class.path" />

</javac>

</target>

<!-- Run the JUnit Tests -->

<!-- Output is XML, could also be plain-->

<target name="junit" depends="compile">

<junit printsummary="on" fork="true" haltonfailure="false">

<classpath refid="junit.class.path" />

<formatter type="xml" />

<batchtest todir="${test.report.dir}">

<fileset dir="${src.dir}">

<include name="\*\*/\*Test\*.java" />

</fileset>

</batchtest>

</junit>

</target>

<target name="main" depends="compile, junit">

<description>Main target</description>

</target>

</project>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LabJUnitAnt-8-Rathna\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Addition.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.demo2.util;

/\*Comments\*/

public class Addition {

public static int twoValues(int x, int y)

{

return x + y;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*Multiply.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.demo2.util;

public class Multiply

{

public static int twoValues(int x, int y)

{

return x \* y;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*Addition\_test.java\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.demo2.util.test;

import com.igate.demo2.util.Addition;

import junit.framework.TestCase;

public class Addition\_test extends TestCase

{

private int x = 0;

private int y = 0;

protected void setUp()

{

x = 4;

y = 5;

}

protected void tearDown()

{

x = 0;

y = 0;

}

public void testAddition()

{

System.out.println("Test use of Addition class");

int z = Addition.twoValues(x,y);

System.out.println(" Result: " + z);

assertEquals(11,z); // The test }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*MultiplyTest.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.demo2.util.test;

import com.igate.demo2.util.Multiply;

import junit.framework.TestCase;

public class Multiply\_test extends TestCase

{

private int x = 0;

private int y = 0;

protected void setUp()

{

x = 4;

y = 5;

}

protected void tearDown()

{

x = 0;

y = 0;

}

public void testMultiply()

{

System.out.println("Test use of Multiply class");

int z = Multiply.twoValues(x, y);

System.out.println(" Result: " + z);

assertEquals(20, z);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*build.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<project name="LabJUnitAnt" default="all" basedir=".">

<description>Buildsd, tests, and runs projx</description>

<property name="build.dir" value="." />

<property name="src.dir" location="${build.dir}/src" />

<property name="test.dir" location="${build.dir}/src" />

<property name="test.htmlreports.dir" location="${build.dir}/test/htmlreports" />

<property name="test.data.dir" location="${build.dir}/test/data" />

<property name="junit.class.name" value="com.igate.demo2.junit.AllTests" />

<!-- Classpath to find java packages -->

<path id="classpath.base">

</path>

<!-- Classpath for tests to find src packages -->

<path id="classpath.src">

<pathelement location="src" />

</path>

<path id="classpath.junittest">

<path refid="classpath.base" />

<pathelement location="${build.dir}/lib/junit-4.4.jar"/>

<pathelement location="${test.dir}" />

</path>

<target name="clean" description="Remove all .class files">

<delete includeEmptyDirs="true" failonerror="false">

<fileset dir="${src.dir}">

<include name="\*\*/\*.class" />

</fileset>

</delete>

</target>

<target name="compile" depends="clean">

<!--<javac srcdir="${src.dir}" destdir="${src.dir}" debug="true"

includeAntRuntime="true">-->

<javac srcdir="${src.dir}" destdir="${src.dir}" debug="true">

<classpath refid="classpath.base" />

<classpath refid="classpath.junittest" />

<include name="\*\*/\*.java" />

</javac>

</target>

<target name="compile-test" depends="compile">

<javac srcdir="${test.dir}" destdir="${test.dir}" debug="true"

includes="${src.dir}" includeAntRuntime="true">

<classpath refid="classpath.base" />

<classpath refid="classpath.src" />

<classpath refid="classpath.junittest" />

<include name="\*\*/\*.java" />

</javac>

</target>

<target name="test" depends="compile-test">

<mkdir dir="${test.data.dir}" />

<mkdir dir="${test.htmlreports.dir}" />

<junit fork="no" haltonfailure="no" showoutput="yes"

printsummary="true">

<classpath refid="classpath.base" />

<classpath refid="classpath.src" />

<classpath refid="classpath.junittest" />

<test name="${junit.class.name}" todir="${test.data.dir}" />

<formatter type="brief" usefile="false" />

<formatter type="xml" />

</junit>

<junitreport todir="${test.htmlreports.dir}">

<fileset dir="${test.data.dir}">

<include name="TEST-\*.xml" />

</fileset>

<report format="frames" todir="${test.htmlreports.dir}" />

</junitreport>

</target>

<target name="all" depends="test" />

</project>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*