**Computer Networks 1**

**Lab 3**

**Java Programming using Eclipse**

**I. Objectives**

1. Practise with basic Java programming

2. Client/Server socket programming

Learning outcomes:

– Know how to use basic programming constructs of Java using Eclipse.

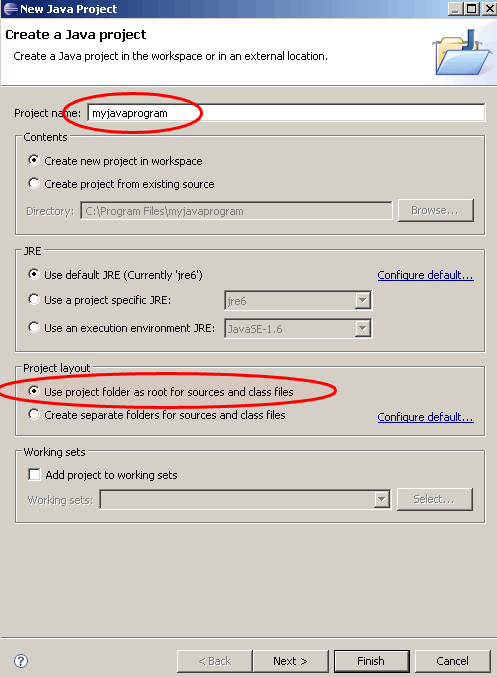
– Able to create TCP sockets for client and server sides in Java.

**II. Programming environment**

1. Eclipse for Java

**III. Contents**

**1. Basic Java programming**



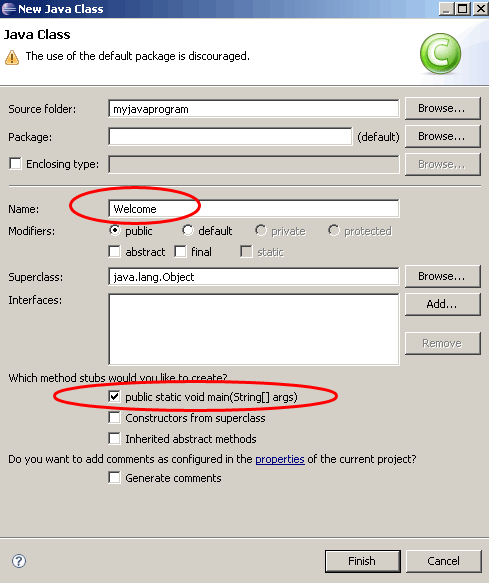
**a. Setting up Java programming environment on Eclipse**

• **Create new project**

– Start Eclipse

– Choose: File -> New -> Java project -> myjavaprogram -> Finish

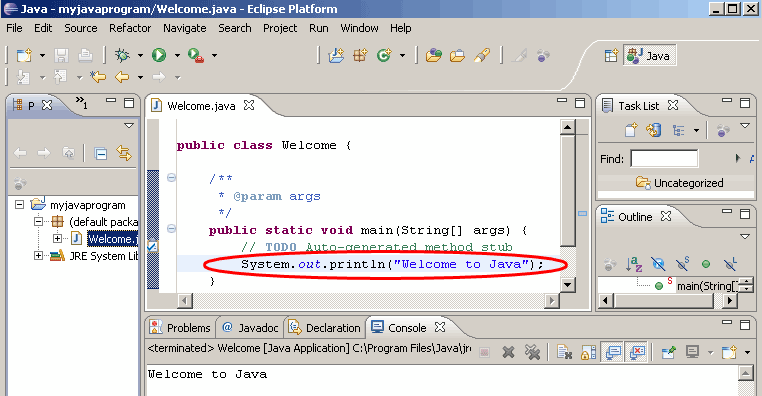
– Choose: File -> New -> Class, type in Welcome, select option *public static void main(String[] args)* -> Finish



You will see the editing screen as below, then:

– Type in ***System.out.println(***"***Welcome to Java***"***);*** in main()

– Choose Compile -> Run



**b. Practise with Java**

• **Print out IP of localhost**

import java.net.\*;

public class HostInfo {

public static void main(String args[]) {

HostInfo host = new HostInfo();

host.init();

}

public void init() {

try {

InetAddress myHost = InetAddress.getLocalHost(); System.out.println(myHost.getHostAddress()); System.out.println(myHost.getHostName());

}

catch (UnknownHostException ex) {

System.err.println("Cannot find local host");

}

}

• **Print out IP of** [***www.hcmut.edu.vn***](http://www.hcmut.edu.vn)

import java.net.\*;

class Info {

public static void main (String args[]) {

try {

InetAddress[] addresses =InetAddress.getAllByName("www.hcmut.edu.vn");

for (int i = 0; i < addresses.length; i++)

{

System.out.println(addresses[i]);

}

}

catch (UnknownHostException e) {

[System.out.println("Could](http://www.hcmut.edu.vn) not findwww.hcmut.edu.vn");

}

}

}

**2. Create server/client sockets**

• **Create a TCP client socket and connect to a server**

import java.net.\*;

import java.io.\*;

public class getSocketInfo {

public static void main(String[] args) {

for (int i=0; i<args.length; i++){

try {

Socket theSocket = new Socket(args[i], 80); System.out.println("Connected to "

+ theSocket.getInetAddress() +" on port "

+ theSocket.getPort() + "from port "

+ theSocket.getLocalPort() + " of "

+ theSocket.getLocalAddress());

}

catch (UnknownHostException e) {

System.err.println("I can't find " + args[i]);

}

catch (SocketException e) { System.err.println("Could not connect to "

+ args[i]);

}

catch (IOException e) { System.err.println(e);

}

} // end for

} // end main

}// end getSocketInfo

• **DateTime Server**

import java.net.\*; import java.io.\*; import java.util.Date;

public class DayTimeServer {

public final static int daytimePort = 5000;

public static void main(String[] args) {

ServerSocket theServer; Socket theConnection; PrintStream p;

try {

theServer = new ServerSocket(daytimePort);

while (true) {

theConnection = theServer.accept();

p = new PrintStream(theConnection.getOutputStream());

p.println(new Date()); theConnection.close(); theServer.close();

}

}

catch (IOException e) {

System.err.println(e);

}

}

}

• **Exercise**

– Write two programs Server.java and Client.java: create sockets to connect client and server together, then assign suitable IP and Port for the sockets.

– Requirements: after compilation, run the Server first, then run the Client to connect the client with the Server.