

```

integer sensorRange = 30;
float sensorFrequency = 10.0;
integer listenChannel = 1000;
// Delay for clearing out latestVisitors list (2 hours in seconds)
float welcomeBackDelay = 7200.0;
list visitors;
list latestVisitors;

default
{
    state_entry()
    {
        llSensorRepeat("", "", AGENT, sensorRange, 2*PI, sensorFrequency);
        llSetTimerEvent(welcomeBackDelay);
        llListen(listenChannel, "", NULL_KEY, "");
    }

    sensor(integer total_number)
    {
        integer i = 0;
        while(total_number > i)
        {
            string userName = llDetectedName(i);
            list nameCheck = [userName];
            integer test1 = llListFindList(visitors, nameCheck);
            if(test1 == -1)
            {
                llDialog(llDetectedKey(i), "Welcome " + userName + "
to the Skill Mastery Hyperdome. Note: Click on the Welcome sign to re-
open this dialog\n** Options for Information **\n1. Notecard: Read and
save to your SL Inventory\n2. Resources: View teacher resources on
WikiEducator\n3. Notes: View student notes on WikiEducator",
["Notecard", "Resources", "Notes"], listenChannel);
                visitors = visitors + userName;
                latestVisitors = latestVisitors + userName;
            }
            else
            {
                integer test2 = llListFindList(latestVisitors,
nameCheck);
                if(test2 == -1)
                {
                    llInstantMessage(llDetectedKey(i), "Welcome back
to the Skill Mastery Hyperdome " + userName + ". To review general
information on this build please click on the Welcome sign.");
                    latestVisitors = latestVisitors + userName;
                }
            }
            i++;
        }
    }

    touch_start(integer num_detected)
    {
        llDialog(llDetectedKey(0), "Welcome " + llDetectedName(0) + "
to the Skill Mastery Hyperdome Note: Click on the Welcome sign to re-

```

```

open this dialog\n** Options for Information **\n1. Notecard: Read and
save to your SL Inventory\n2. Resources: View teacher resources on
WikiEducator\n3. Notes: View student notes on WikiEducator",
["Notecard", "Resources", "Notes"], listenChannel);
}

listen(integer channel, string name, key id, string message)
{
    if(message == "Notecard")
    {
        llGiveInventory(id, llGetInventoryName(INVENTORY_NOTECARD,
0));
    }
    else if(message == "Resources")
    {
        llLoadURL(id, "View teacher resources on WikiEducator",
"http://wikieducator.org/Foundation_interview_skills_in_Second_Life/Te
acher_resources");
    }
    else if(message == "Notes")
    {
        llLoadURL(id, "View student notes for interviews on
WikiEducator",
"http://wikieducator.org/Foundation_interview_skills_in_Second_Life");
    }
    else if(message == "say list" && id == llGetOwner())
    {
        integer i;
        for (i=0; i<llGetListLength(visitors); i++)
        {
            llOwnerSay(llList2String(visitors, i));
        }
    }
}

timer()
{
    // Remove oldest half of latest visitors list
    integer halfLen = llGetListLength(latestVisitors)/2;
    latestVisitors = llDeleteSubList(latestVisitors, 0, halfLen);
    // If visitors list over 80 remove oldest visitors from list
    if(llGetListLength(visitors) > 80)
    {
        integer over80 = llGetListLength(visitors) - 60;
        visitors = llDeleteSubList(visitors, 0, over80);
    }
}
}

```