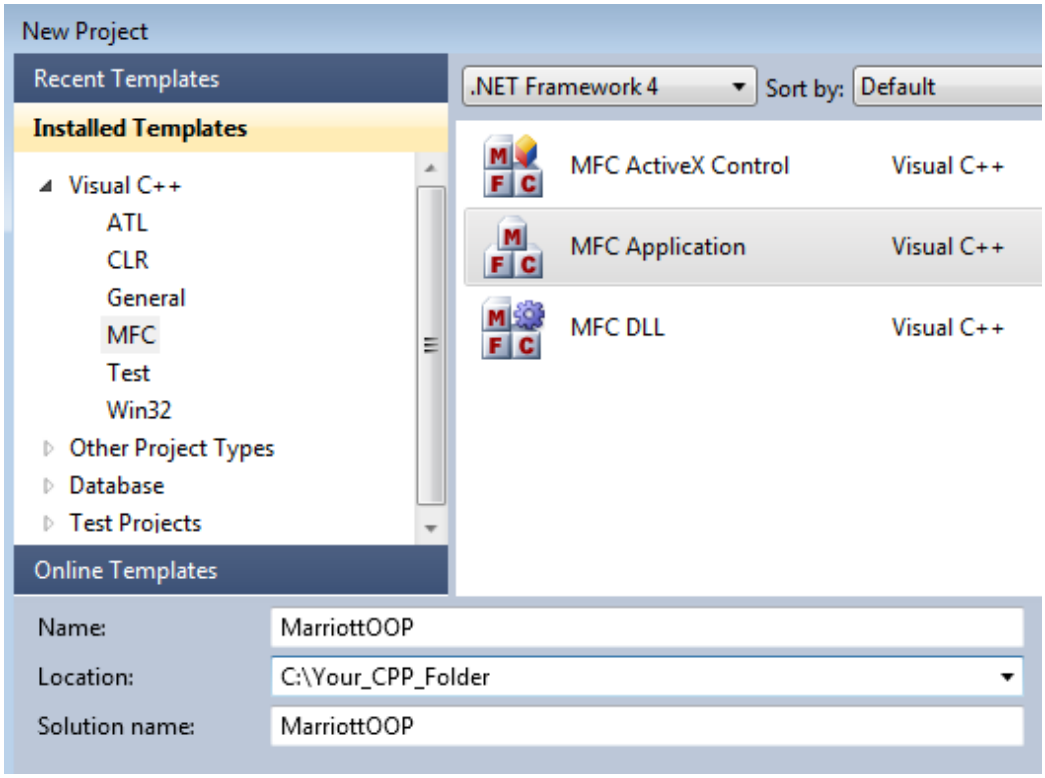
.1) Your 1st Windows Program in the Visual C++.NET Environment ::

Run Microsoft Visual C++.NET

File...
New...
Project…
Visual C++ …
MFC…
MFC Application



Set ***Name*** to "Your Last Name"OOP (Ie MarriottOOP)

Change ***Location*** to an appropriate spot. Click ***OK***.

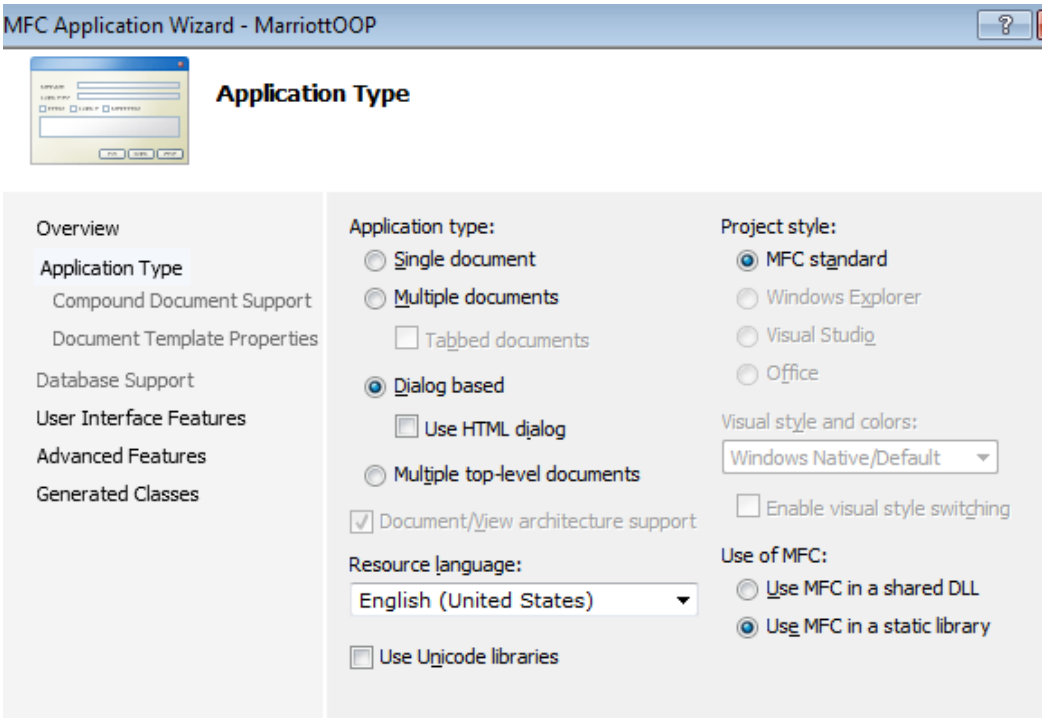
Make sure you select

Application Type…
Dialog Based

Use MFC in a static library

Uncheck
Use Unicode libraries

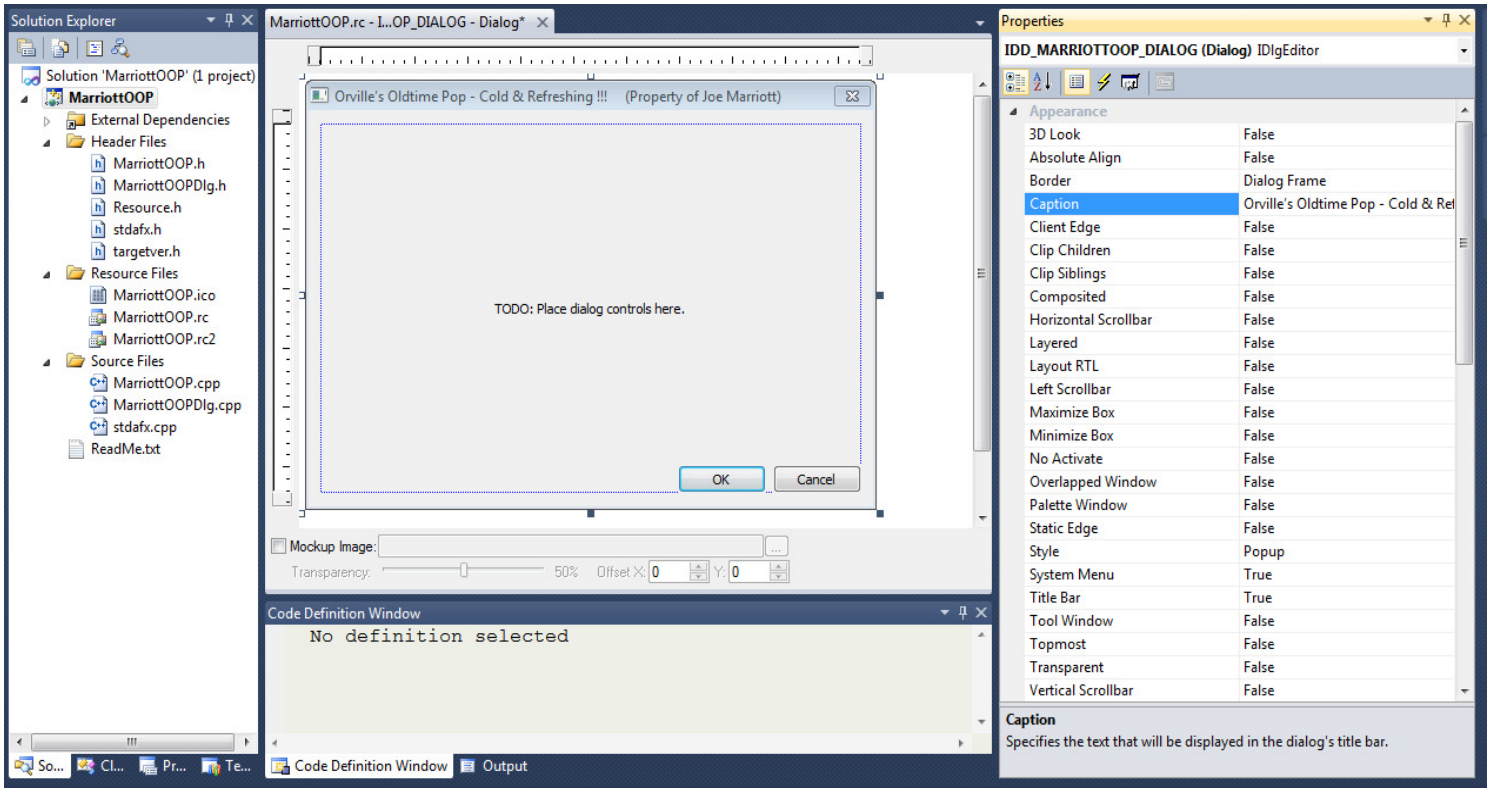
Click
Finish



01.1201.OOP1 :: Orville's Oldtime Pop (OOP) :: Phase 1

Click on your Dialog to select it. Locate ***Caption*** in the Properties tab , and change it to

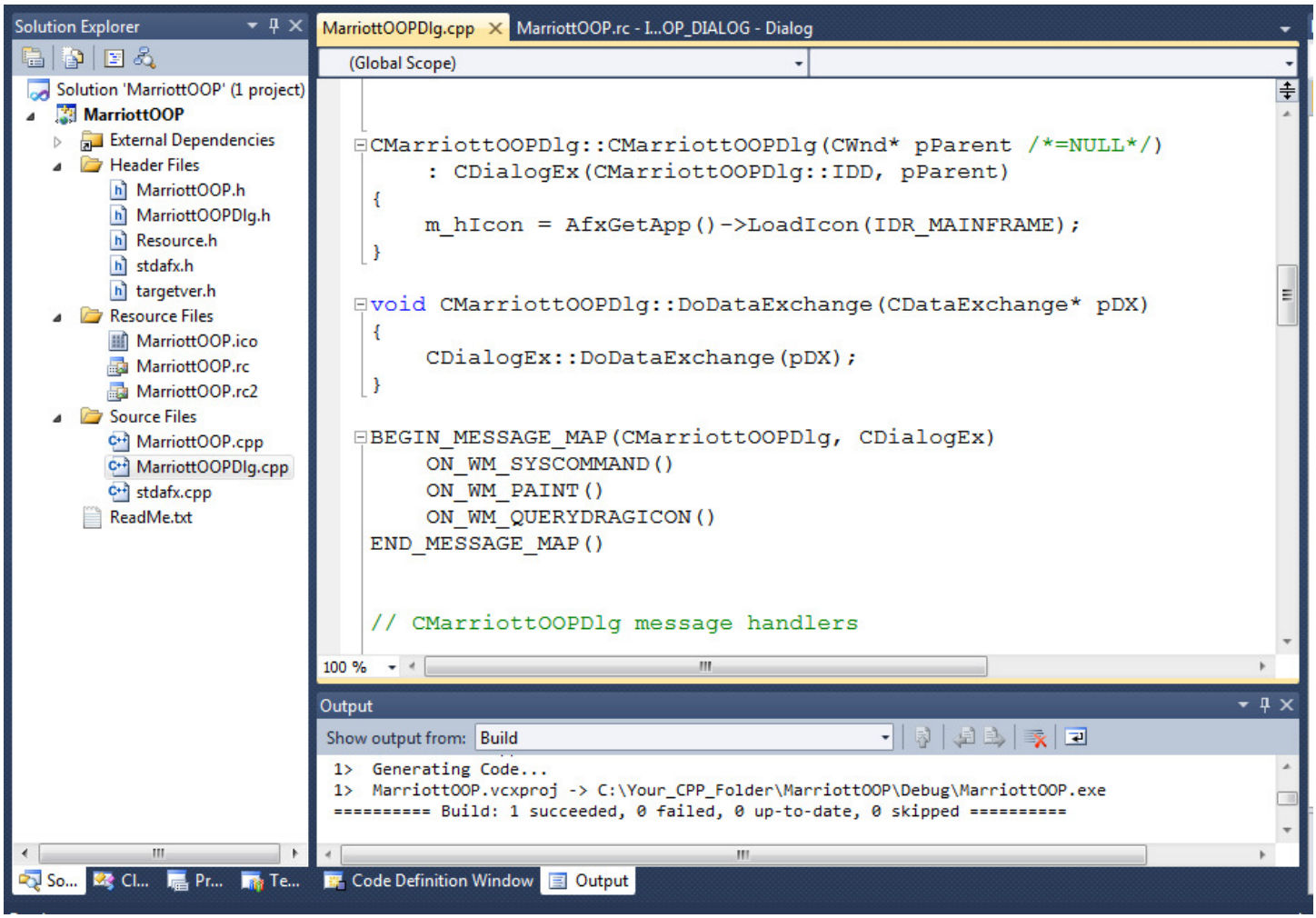
Orville's Oldtime Pop - Cold & Refreshing !!! (Property of <YOUR NAME>)



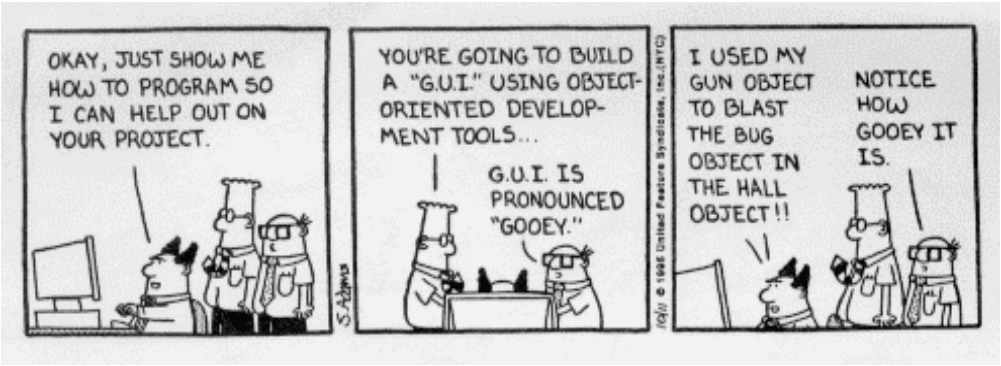
Now, ***Build... Build Solution ,*** and then ***Debug… Start .***

If you don't have any errors, you should soon have your first Visual C++.NET Windows program completed! Hurray!

OK, maybe you're not impressed. BUT, NOW TAKE THE TIME to explore everything that has been generated to accomplish that! Look at everything under ***Solution Explorer***, ***ClassView***, and ***ResourceView***.



Feel free to open the source files under ***Solution Explorer*** (as long as you're careful, you shouldn't hurt anything!).



Impressed now? Think of programming that!!!

Now... we just have to get it to do something more substantial...

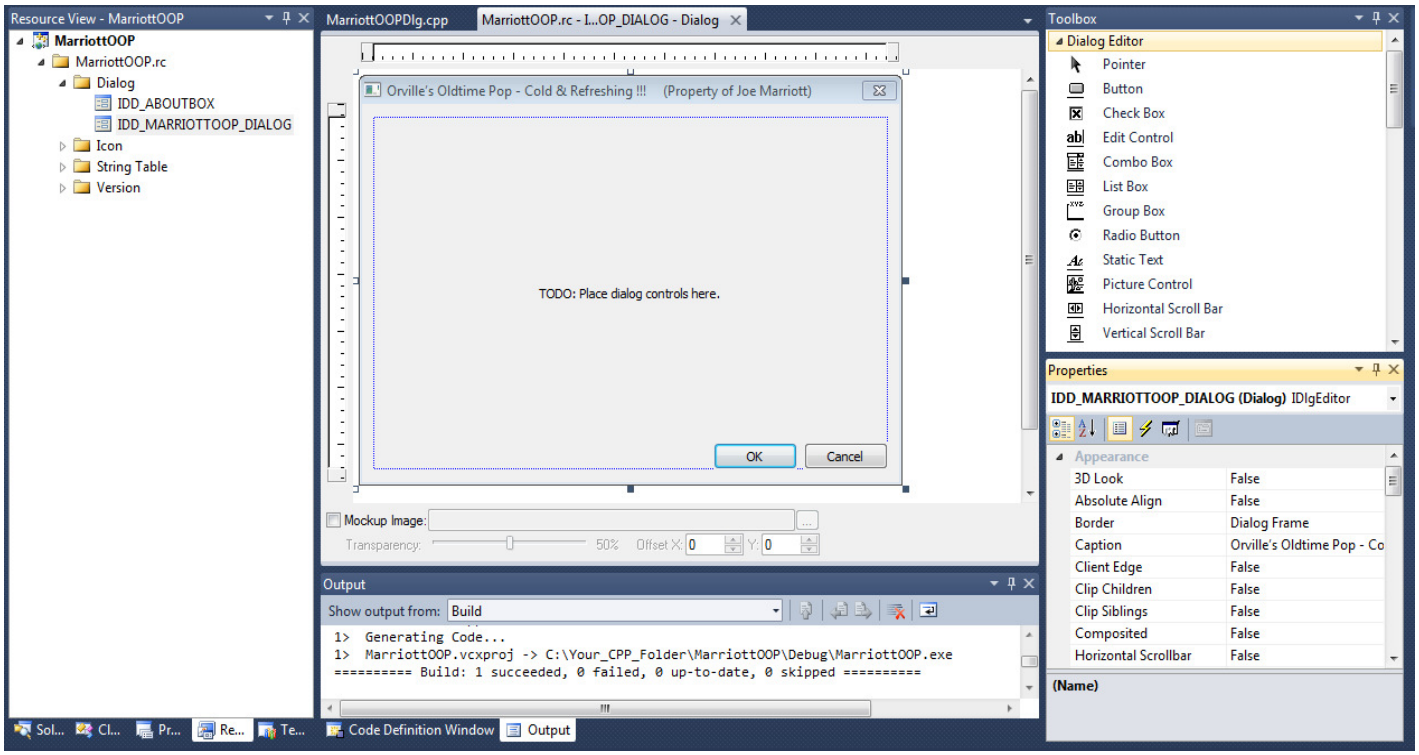
.2) Designing your Screen ::

Before proceeding further, it is assumed that you have spent time using the provided working model of this project (**oop.exe** found in the course folder), and you fully understand the mechanics of what we are aiming for!

OK, let's do the fun stuff first! And, since this is a "visual" program, we might as well have the full screen created.

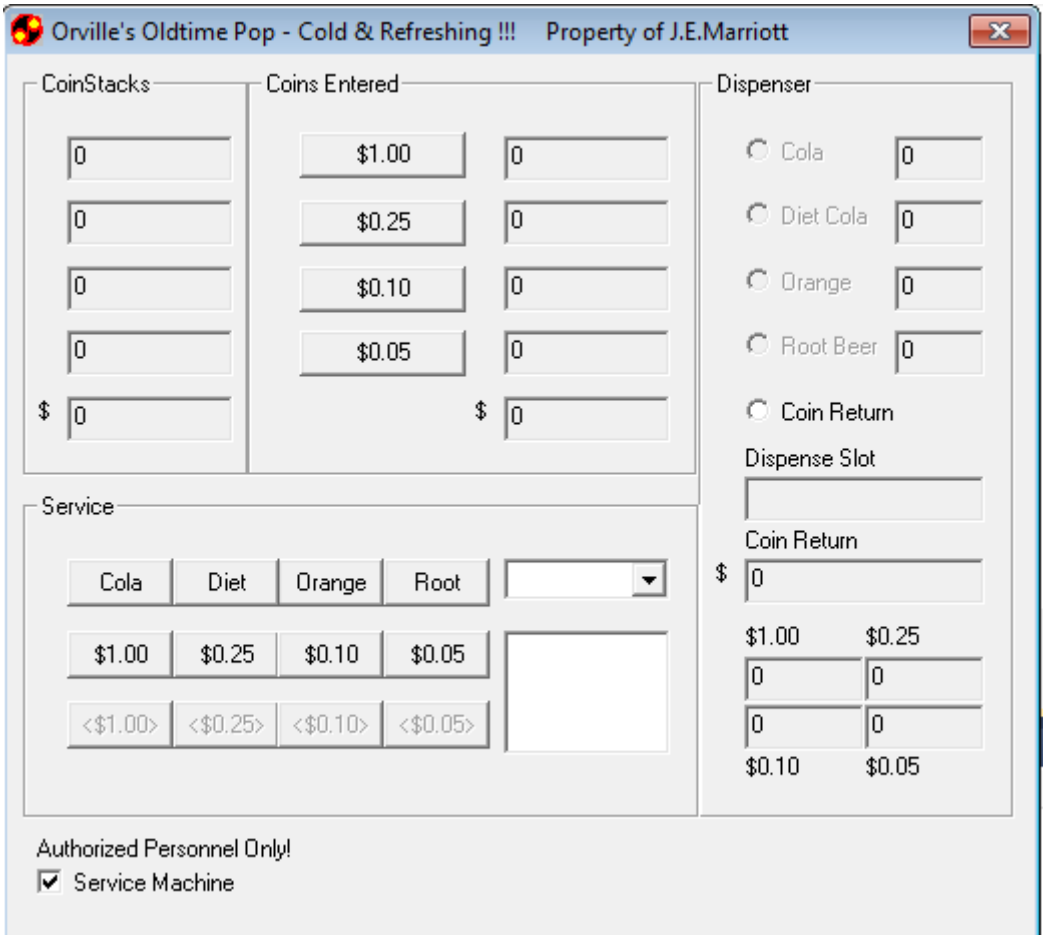
I know you have experience with other visual development environments. So, it should be interesting to see how that knowledge applies to this environment (I'm predicting it's pretty much transferable).

Click on the ***ResourceView*** tab, expand the ***Dialog*** folder, and open **IDD_???OOP_DIALOG** (where ??? should be your last name)



There you go. Have fun. The screen layout you are designing is found on the next page…

01.1201.OOP1 :: Orville's Oldtime Pop (OOP) :: Phase 1



As much as possible, you are to create a carbon copy!!! A screen design which has been obviously thrown together with no time given to control sizes, placement, and alignment will be considered non-compliant in this course (and others, hint) and will receive appropriate evaluation.

Points of interest... you should have access to a Toolbox of controls, and if you look along the top you see a series of icons designed to assist you in sizing/positioning controls. As well, commands are also available in the ***Format...*** menu selection at the top of the screen.

How do you learn how to do this?... experience!!! Jump in... it gets easier the more you do it!

I've provided a full working model of what you're aiming for, but there's a hardcopy representation for you to work from as well. You can find it near the end of this document.

NOTE!!!

DO NOT ATTEMPT to set the labels of the controls (ie $1.00, Cola, CoinReturn, etc. etc.)

THAT WILL BE DONE in the next step!!!

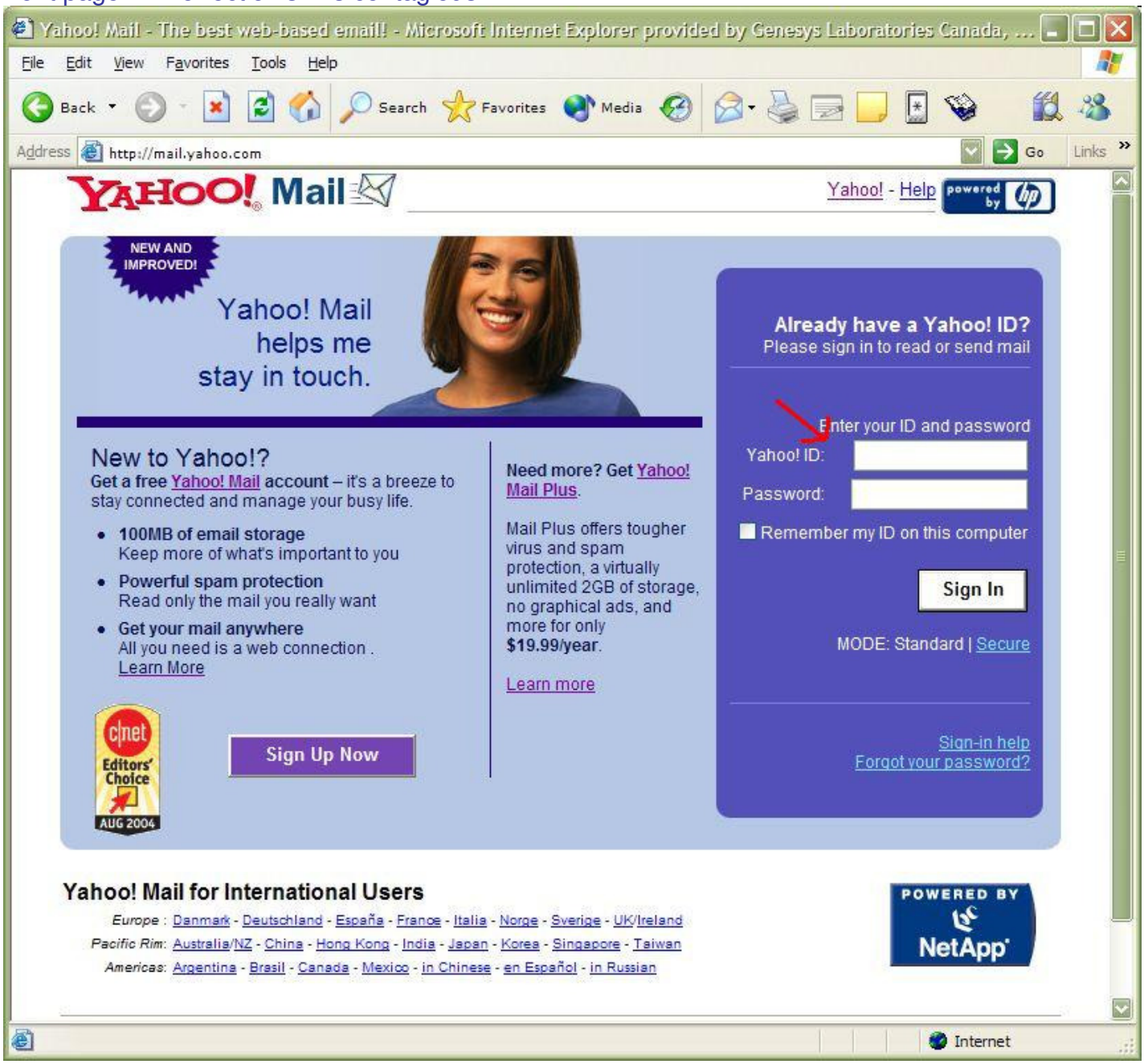
For now, just get comfortable with using the design tools!

.3) Designing your Screen :: Details? Quality?

You're training to design professional products. Quality is Job 1 !!! Consumers are becoming more and more concerned about the lack of quality in software products.

But don't believe me. Here's a former student. I think she's a convert…

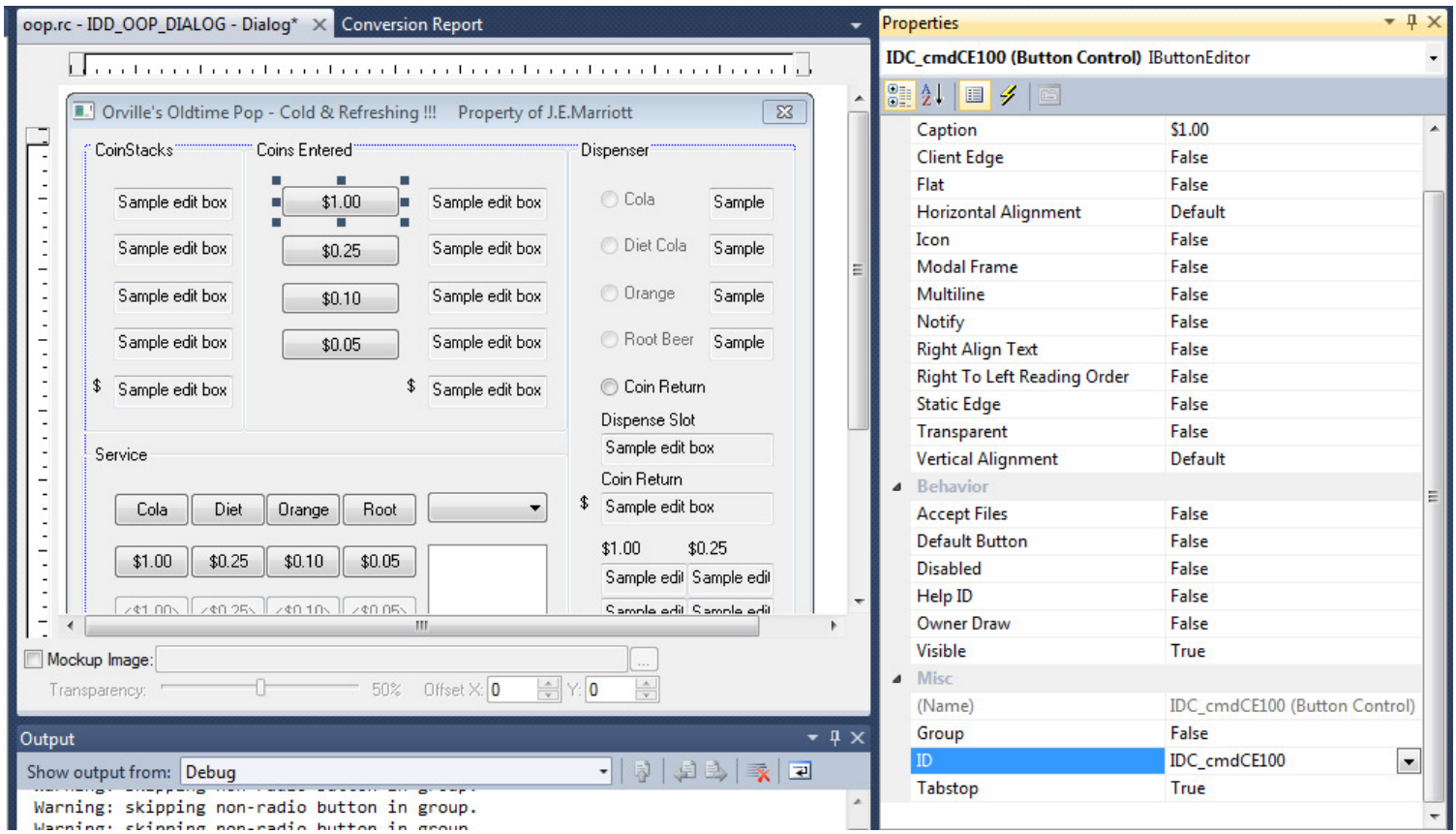
"It's your fault Joe… .....that this drives me crazy EVERY time I login to Yahoo mail. Who QA'ed this front page?? Perfectionism is contagious."



So, is it important what your screen design looks like? Is control alignment important? Spelling? Consistency in design elements? Try answering YES to all of these with what you submit! ☺

.4) Labelling and IDing your Controls :: The Paperwork

If you completed the previous task correctly, your screen should consist of 45 or so different elements. Static text controls are not listed here (yet).



Each control needs a UNIQUE ID so that Windows knows "who's being poked". Click on a control, and then under ***Properties***, you'll see that the environment has already set these for you. Unfortunately, in two weeks when you're up to your elbows in code, do you REALLY THINK you'll remember what **IDC_EDIT1** maps to? Ok. How about **IDC_EDIT13???** NOT!

So, time to do some paperwork to get properly prepared for the journey ahead. You have been provided with two other documents... 1.Appendix1.Screen Layout and 1.Appendix2.Control Ids, Assigned Variables. Get out your pen!...

On the Screen Layout, number each control (1, 2, 3, etc). It doesn't matter what order you do them in, but I'm assuming you have some sense of organization. Make sure everything you "dragged" onto the screen has a number (including the frames and static text).

Next, on the Control IDs form, fill in the column corresponding with each field number.

I was going to give you all the ID's but remembering where you are in this organization, I figured that would be too much spoon feeding. You should know the value of appropriate names by now, and if you don't, the magnitude of this environment will get the point across once and for all!!!

All ID names should start with **IDC_** !!! , followed by the Windows Standard Object Naming Convention...

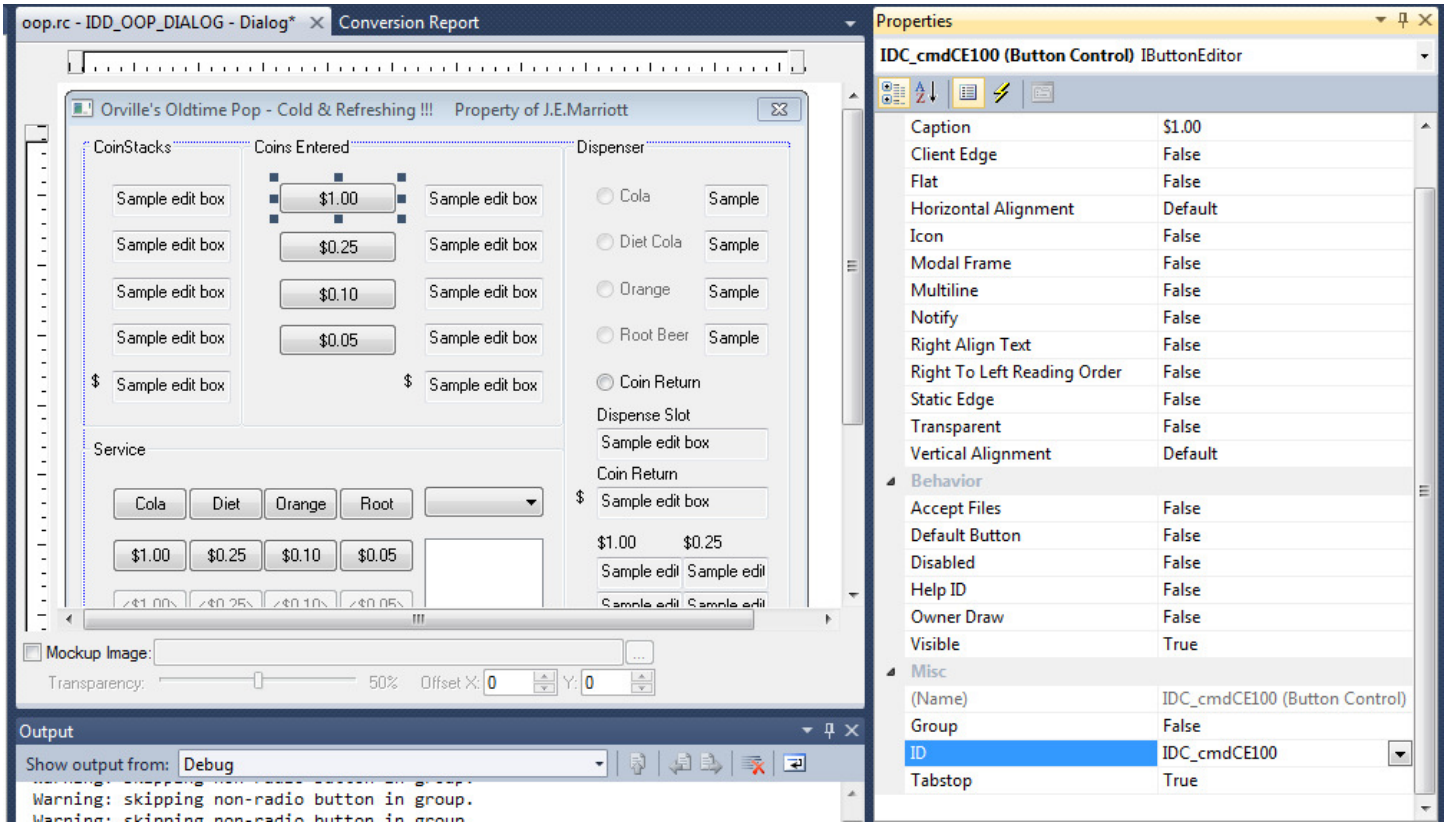
Check Box	chk	chkSend	Image	img	imgFlower
Combo Box	cbo	cboLanguage	List Box	lst	lstProv
Command Button	cmd	cmdExit	Menu	mnu	mnuLoad
Data Control	dat	datAccount	Option Button	opt	optAMFM
Form	frm	frmMyForm	Text Box	txt	txtMember
Group/Frame	grp	grpButtons	Vertical Scroll Bar	vsb	vsbAccNum
Horizontal Scroll Bar	hsb	hsbProperties			

, followed by a descriptive name. OK, need a little hint?... I'd call the $1.00 Button **IDC_cmdEnter100**, the Cola Radio Button **IDC_optSelectCola**, etc.

Try to make your control names mimic the actual action of the Machine Control/Action!

For now, don't worry about the column "**Variable (if any)**".

.5) Labelling and IDing your Controls :: Implementing



Chose a control on your screen, and click on it. In ***Properties***, you'll be able to change the ***ID*** and ***Caption*** for that control. OK, now do all of them... that should keep you busy for awhile!

When completed, rebuild and run your program. Although it still has no true functionality (just a "few" things left to do!), your model should be a mirror image of the provided model.

.Z) Checkpoint ::

At this point, you should be comfortable with generating a Dialog Based Windows project with a complex screen design. As well, you should be familiar with how to assign (or view) the IDs of the various controls.