THE UNIVERSITY OF NEWCASTLE AUSTRALIA

School of Electrical Engineering and Computing

Inft2012 Application Programming – Notes for week 3

Note on writing and running programs

Sometimes when you click the Start debugging button in Visual Studio you get a message that says "There were build errors. Would you like to continue and run the last successful build?" *There is never any point in answering Yes to this question!*

Build errors are what we call syntax errors. Visual Studio is telling you that there are errors in this program and it can't be run. However, if you want, it's prepared to find and run the last error-free version you had.

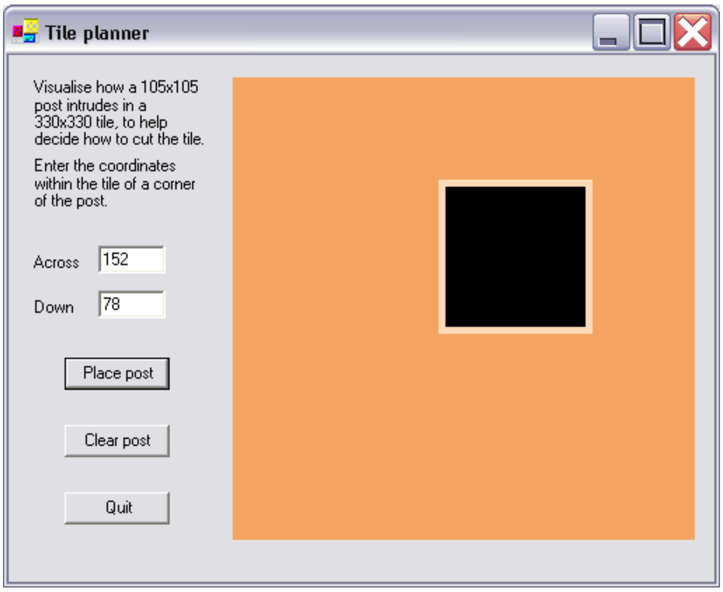
Since you ran that last version you've made changes. The last error-free version was created before you made those changes, so it doesn't show you what the current version does; in fact the current version does nothing, because it has syntax errors and can't be run.

What you have to do is find and fix the syntax errors, then run the newly corrected program.

Lab exercises

Most students will probably not finish all of these exercises in the lab class. Remembering that learning to program takes lots of practice, you are strongly advised to finish them before next week's lab class, bringing to the class any problems you encounter.

1. You are tiling a deck with tiles that are 330x330mm in size. At various points on the deck there are posts 100x100mm in size. Write a program to visualise where each post fits into its surrounding tile(s), to help you decide how to cut the tile. The interface will look like this . . .



Tile planner

Visualise how a 105x105 post intrudes in a 330x330 tile, to help decide how to cut the tile.

Enter the coordinates within the tile of a corner of the post.

Across 152

Down 78

The pale area around the post is a 5mm allowance for grout, the cement-like filler used between tiles. Write the program. When designing it, decide whether you need any methods other than event handlers.

Remember to ensure that the picture box has a SizeMode of Normal; otherwise the images might be distorted.

Deliberately create an accidental handler by double-clicking on the form, a label, the picture box, or a text box. See how to remove it immediately by going back to the Design mode and selecting Undo. Now create another, select it in the code window and delete it, and proceed to find and delete the troublesome line of code as described in the lecture.

2. Adapt the Aboriginal flag program (Lecture3DemoFlags) so that it draws 'targets' of different sizes in different locations. A target is made up of three concentric circles, one colour on the outside, then a second, then a third in the middle. Make each of the outer rings the same width (not diameter), and give the inner circle a radius equal to that width.

3. If a solid cylinder has height h and radius r, its surface is made up of two circles of radius r and a (curved) rectangle whose length is h and whose breadth is the circumference of the base. A circle of radius r has an area of πr^2 and a circumference of $2\pi r$. Write a function method that returns the surface area of a solid cylinder, given suitable arguments. Test your method in a suitable program.

4. Thoroughly examine your die-drawing program from last week. There's probably a lot of repetition in it, which could be greatly reduced by writing appropriate methods. Find the repetition, write the methods, and see if you agree that this improves the program. Now add a button labelled "Roll". When the Roll button is pressed, your program should use the Random class to generate an appropriate number, and then display the corresponding die.

5. Your body mass index (BMI) is your mass in kilograms divided by the square of your height in metres. Write a program that accepts these two measurements, and when a button is pressed, calculates the BMI and displays the BMI and a suitable message. A BMI less than 20 suggests that a person is underweight; 20-25 is in the healthy weight range; 25-30 is overweight; and more than 30 is obese. Simon's not overweight – he's just seriously underheight!

6. This truly horrible piece of code illustrates holes in scope. Desk check the code to see what the values of all the variables will be as the code executes. Confirm your desk checking by running the code, using message boxes or some other means to display the values at appropriate points of the program.

```
int i1, i2, i3;

private void Example()
  {
       i1 = 1;
       i2 = 5;
       i3 = PlayWith(i1);
  }

private int PlayWith(int i2)
  {
       int i3 = i1 + i2;
       i1 = i2 + i3;
       i2 = 23;
       return i2 - i1 + i3;
  }
```