CPS606
Spring 1998

Homework Guidelines

Here are some guidelines for preparing your assignments. We plan to follow them in the grading starting from Homework 3. **Description**

1. You need to give a short definition of your application. For example, if you are trying to design, let say, an Airline Reservation System, then you need to tell us what an airline reservation system is and also the methods you use to solve the problem.

2. Explain clearly the flow of data within your program, where it originates, where it ends, for example:
   "The user enters the data or read-in using function(x). Then it gets processed by program or function(y), etc. Function(z) handles the output and the database, etc. ..." or something similar.

   An example diagram:

   ```
   +------+   data   +-----------------------+ output  +----+
   | user |--------->| program or subroutine |-------->| DB |
   +------+          +-----------------------+         +----+
      ^      response          |
      +--<--<------------------+
   ```

   Note: Having a diagram only is not sufficient.

   Program flow should also be described. This could include a brief overview of classes, threads, or events that your program will respond to.

3. Hyperlink all the references you used. For example, if you have used a method that was written or suggested by someone else, then, if possible, you need to provide a link to the page where you got it from. In addition, if you only partially have adapted someone else's code, you need to briefly tell us that.

   Comments in your programs

4. In the source code you will need to have a title and a brief description of what each program or class does. Also, you need to have a short title for each class or method. Inside these titles you will briefly describe what the method does, and whether you have copied it from somewhere else or have written it yourself. This point is very important.

5. Also at the top of each class or method, you will need to define your parameters and return values (if any) as well as your local variables.

6. Modularization is quite important so try not to cram too many things in one method. If it gets

really big, either divide it or have clear comments between the source blocks, i.e. before and after whiles, ifs, trys, etc.