ESC101 : Fundamental of computing

Tutorial sheet 6 **18 September, 2008**

Topics covered : Only two classes were held since the last tutorial (the class on 15 September could not be held due to some problem). I considered a few examples for problem solving. After that I announced that we shall discuss another programming methodology : Object oriented programming. The material on OOP which I covered is mentioned in the beginning of second part of the sheet below and should be discussed in the second half of the tutorial. The lecture notes (lec19.pdf and lec20.pdf) are available on the course website for your references.

1. (**Problem solving**)
   Solve the following problem interactively. Given a positive integer n find the largest positive integer x such that

   $$n = m^x \quad \text{for some integer } m > 0$$

   The students are not allowed to use any library function. There is a trivial solution for the above problem using two nested `for` loops and doing exhaustive search for m and/or x. Another solution is based on the following observation : Let $p_1, ...p_k$ be the prime factors of n, then

   $$n = p_1^{a_1} \times p_2^{a_2} \times ... \times p_k^{a_k}$$

   With this expression, it follows that

   $$x = GCD(a_1, ..., a_k) \quad \text{and} \quad m = p_1^{a_1/x} \times p_2^{a_2/x} \times ... \times p_k^{a_k/x}$$

   How will you solve the given problem based on above approach. What methods will you design ? Please do not use any arrays here.
   One possible solution using second approach : Run a for loop to compute GCD of a_i's while also computing p_i's implicitly. Also discuss if there is any other way to solve the problem.

2. (**Discussion and clarification of doubts in OOP**)

   - I announced in the class last Friday that we shall cover object oriented methodology in next few weeks and that this new methodology is different from structured programming. I gave a small historical perspective of various programming methodologies.
   - I have covered the following concepts with some examples (You may consult Today's lecture lec20.pdf) :
     Encapsulation
     Object
     Class
   - Please clarify the difference between an object and its class with examples. Though I used many examples to show the difference, this important concept will take some time to sink in the minds of the students.
   - Based on the questions asked by few students after the class I think the students will have questions in creating objects, the reference to object. Please clarify these doubts. For this you may view a few slides of the lecture 20 held today and explain the last slide in detail again using diagram (with reference as an arrow/pointer to the object).
   - Most importantly, if you could highlight OOP as a very natural and *close to humans* way to solve the computational problems of real life, it would be great. I spent most of today's lecture on this concept. I think it will be still a good idea to explain it again in your own words.

   For the above suggested topics, please cover only as much as you wish. But please go very slow since most of the students are exposed to OOP for the first time.
   Please **DO NOT EXPLAIN** the meaning of **public** or **private** classes/methods and **static** or **dynamic** methods since these will be covered in the lectures soon. Do not discuss methods whose parameter is a reference to some object. This will be covered soon in the lectures.