ESC101 : Fundamental of computing

Tutorial sheet 3 **14August, 2008**

I covered `for` and `while` loop (but I did not discuss `break` and `continue` commands). Please clarify any doubt if they have. Thereafter, please involve the class to solve the following problems interactively.

1. **For loop example**
   The following code is supposed to print the sum of cubes of all those numbers which are in the range [1,500] and are not multiples of 3. Fill in the blanks appropriately.

```
1.class while_example
2.{
3.   public static void main(String args[])
4.   {
5.     int i;
6.     int j;
7.     i=1; j=0;
8.     int sum;
9.     sum=_______________;
10.    while(i<=________)
11.    {
12.       if(____!=____)
13.       {
14.              j=_______;
15.            sum=sum+j;
16.       }
17.       i = ________;
18.    }
19.    System.out.println(``The sum is ''+sum);
20.  }
21.}
```

   Please involve the entire class and proceed in a analytical fashion to fill in the blanks.

2. **Prime number**
   The problem is to find out whether a given number `i` is prime. Starting from the definition of prime, discuss the common way by which one solves this problem with paper and pen : starts from 2, find out if `i` is multiple of 2. If not, find out if `i` is multiple of 3, and so on. Try to point out that this method involves repeating similar step a number of times. So it is an ideal place for using loop to solve the problem. What should be the stopping condition ? Write the program interactively.

   Encourage the students to run this program in cc and to explore what happens if the number is very large (example: 9007199254740989 is prime)? Then encourage them to find out ways to make the program fast (check for a factor only upto $\sqrt{i}$ and/or only odd numbers, ...). Please note that I have not discussed complexity of algorithm till now. I shall discuss many more problems in the class, labs and tutorial so that they realize the importance of efficiency of a program in a very natural manner. Towards the end of this course, I shall introduce the complexity of algorithms.