

Lecture 15: Classes, Inheritance, Access Control

Inheritance and Access Control

```
1 class Chart:
2
3     def __init__(self, title):
4         self._title = title
5
6     def title(self):
7         return self._title
8
9     def __str__(self):
10        return "{}".format(self._title)
11
12 class Histogram(Chart):
13
14     def __init__(self, bins, title):
15         self._bins = bins
16         self._counts = [0]*len(self._bins)
17         super().__init__(title)
18
19     def _index(self, bin):
20         return self._bins.index(bin)
21
22     def add_to_bin(self, bin, count):
23         self._counts[self._index(bin)] += count
24
25     def count(self, bin):
26         self._counts[self._index(bin)]
27
28     def __str__(self):
29         h = " ".join(["{}:{}".format(x,y) for (x,y) in zip(self._bins,self._counts)])
30         return "{} {}".format(super().__str__(), h)
```

Project Euler Problem 12: Highly Divisible Triangle Numbers

The sequence of triangle numbers is generated by adding the natural numbers. So the 7th triangle number would be $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$. The first ten terms would be:

1, 3, 6, 10, 15, 21, 28, 36, 45, 55, . . .

Let us list the factors of the first seven triangle numbers:

1: 1

3: 1,3

6: 1,2,3,6

10: 1,2,5,10

15: 1,3,5,15

21: 1,3,7,21

28: 1,2,4,7,14,28

We can see that 28 is the first triangle number to have over five divisors. What is the value of the first triangle number to have over five hundred divisors?