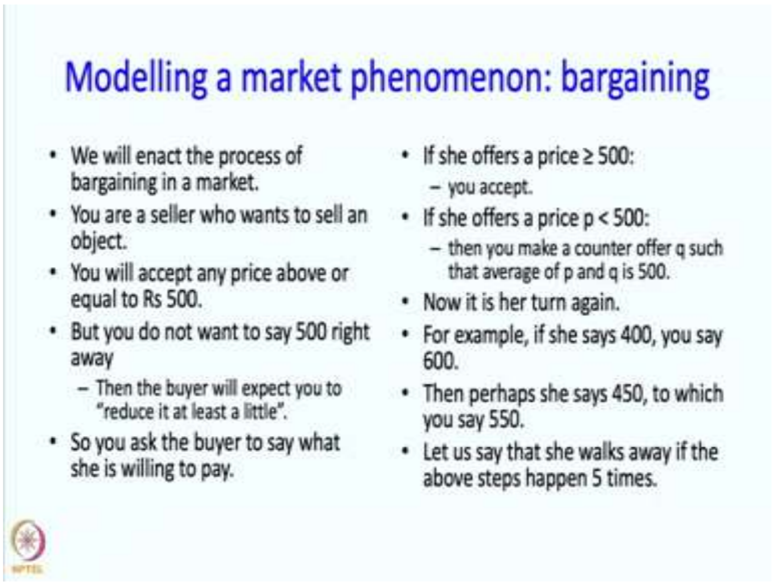
An Introduction to Programming through C++

Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Lecture No. 09 Part - 3
Loops in various applications
Modelling a system: bargaining

(Refer Slide Time: 00:39)



Modelling a market phenomenon: bargaining

- We will enact the process of bargaining in a market.
- You are a seller who wants to sell an object.
- You will accept any price above or equal to Rs 500.
- But you do not want to say 500 right away
  - Then the buyer will expect you to "reduce it at least a little".
- So you ask the buyer to say what she is willing to pay.
- If she offers a price ≥ 500:
  - you accept.
- If she offers a price p < 500:
  - then you make a counter offer q such that average of p and q is 500.
- Now it is her turn again.
- For example, if she says 400, you say 600.
- Then perhaps she says 450, to which you say 550.
- Let us say that she walks away if the above steps happen 5 times.

Welcome back, in the last segment we discussed brute force search for solving problems, for which we do not know anything cleverer. In this segment we are going to see, how to write programs to model the working of a system. We are going to start off with a simple and problem that you might perhaps consider to be a fun problem. Modeling of the bargaining that goes on when we go to a market.

So, pretend that you are a seller who wants to sell a certain object. Now, you will be happy to accept to sell the object for say 500 rupees. So, you will accept 500 as a price and of course, if somebody says on I am offering 600, you will certainly accept it, but you do not, you are not going to accept a smaller price than 500. That is sort of your, that is the minimum price that you want.

And you could say 500, and you could put a tag saying it is 500. But sometimes, in some places the customers want to bargain, so if you just say 500, then the buyers will tell you, oh please reduce, please reduce. And they will not accept it. And if you at accepted may be, you could reduce and you could come down to 500 or something like that. That might be acceptable, so it is just a psychological game, I am sure which you must have seen happening in markets.

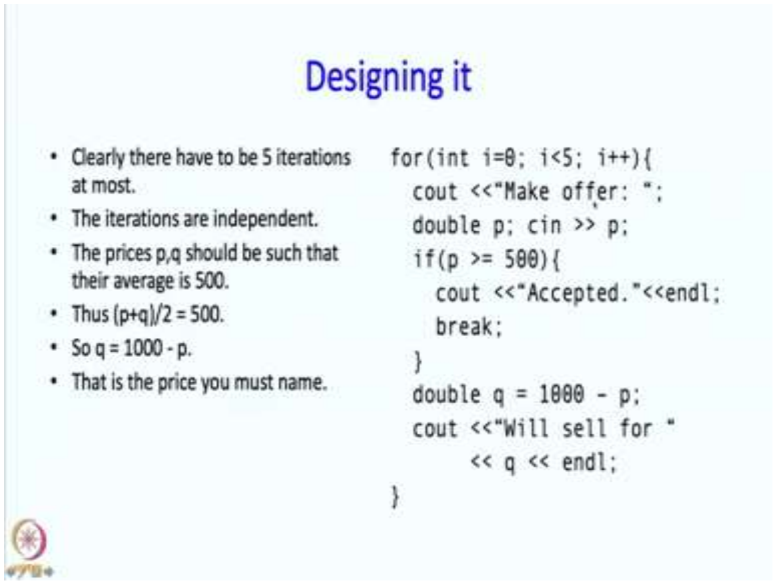
So, how do you deal with this as a seller? Well, so here is the strategy that we are going to employ in this program. So you are not going to tell your price at all. So you are going to say to the buyer, look why do not you tell me, how much you are willing to pay? So if the buyer offers a price bigger than 500, then you are done, you can just take it, you are happy, the buyer are happy, and the whole thing ends over here.

But if, the buyer offers a price smaller than 500, some price p, then what should you say? Well you are going to make, let us say that your strategy is to make a counter offer q, such that the average of p and q is 500. So, once you do that the buyer can accept it, or may be make another offer. And this can go on.

So here are some examples. So suppose, the buyer say look I am willing to pay 400 rupees for it. What are you going to do? Well you will say 600, why is that? Because the average of 600 and 400 is 500. Why the strange thing, why do not you say 500 right away, again you want to keep yourself some leeway in case you need to reduce further. So, this is how you are going to work.

After this may be the buyer says 450, so in this case you will say 550, why? Because again you are sticking to your rule, the average of 550 and 450 is 500. So you must have seen something like this happen, it is not always exactly like this, but let us just say for fun that in fact, well there could be a seller, who uses this strategy. So what you are going to do is? We are going to show, or we are going to write a program to show what is going to happen. If there is a seller like this, and a buyer comes to that seller. And of course this cannot go on at infinitum. So, may be if this happens 5 times then the whole thing has to stop, anyway. So, may be it means that you did not come to an agreement and then, the buyer just walks away. So, this is what you want to your program to enact or simulate.

(Refer Slide Time: 05:02)



Designing it

- Clearly there have to be 5 iterations at most.
- The iterations are independent.
- The prices p,q should be such that their average is 500.
- Thus (p+q)/2 = 500.
- So q = 1000 - p.
- That is the price you must name.

```
for(int i=0; i<5; i++){
  cout <<"Make offer: ";
  double p; cin >> p;
  if(p >= 500){
    cout <<"Accepted."<<endl;
    break;
  }
  double q = 1000 - p;
  cout <<"Will sell for "
      << q << endl;
}
```

So, let us design that program, so clearly there can be at most 5 iterations. And you can see that the iterations are independent there is nothing from the first iteration that is necessary in this second iteration. Now, we have to figure out one thing however, that if the buyer a demands a price p, or rather the buyer offers to pay a price p, then what price q should you make as a counter offer?

Well, we said that p and q should be such that their average is 500. So, what does that mean? So the average of p and q is (p+q)/2, and this must be 500. So, let us simplify that so let us take 2 to the other side and p to the other side, so what do we get? So, we get q must be equal to 1000 minus p. So, this in fact tells you precisely, what your counter offer should be.

So, you could think of this as solving an algebraic equation, so you know the value of p and you are trying to figure out the value of q. So that is exactly what this is, and all I am just alerting you to the possibility that when you write a program, you may need to solve equations, and this is just an example. So do not worry about it, equations can be solved we have shown you, how to do it.

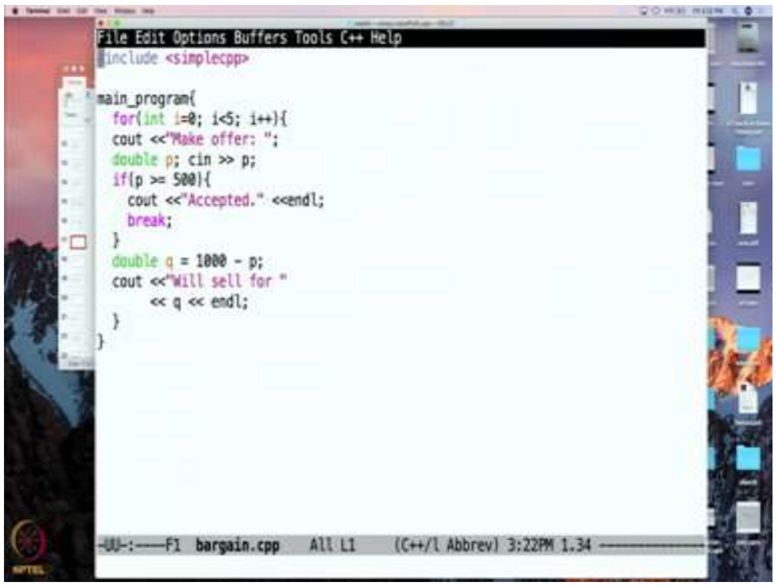
So, q is this price you should name, and that is about it, and of course if the price was the p itself was already above 500, then you should stop. So, how does this is work? So we are going to have a loop which runs 5 times at most, so you are going to tell the user, tell the buyer to make an offer. So, may be the buyer is going to make an offer, if the buyer going to type something and that is going to into p, if p is bigger than or equal to 500, then you are going to accept the offer, so you are going to printout accepted, and you will break.

So, the loop ends, otherwise you have to make a counter offer and how much offer, how much value should that counter offer be for? Well we just calculated that, so we are going to make an offer q equals 1000 minus p. So, you could say I am going to sell it for q, and that is the end of iteration. So, at this point the user, the buyer may make another offer, may be the buyer, so in this precool the buyer is not really allowed to say accepted. The buyer has to come back and make the same offer at which the seller says accepted, but anyway so, if you make an counter offer the buyer can accept at that price by repeating that offer. Alright, so that is the program.

(Refer Slide Time: 08:25)

Demo

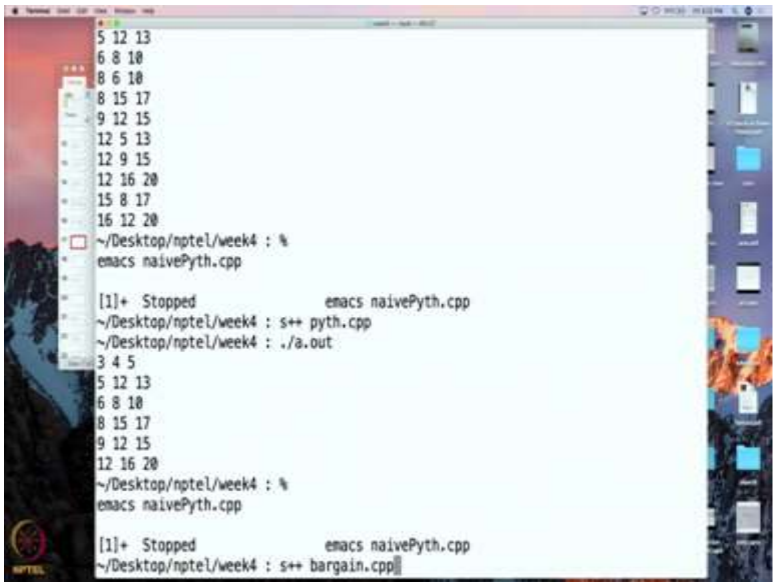
- Bargain.cpp



```
#include <simplecpp>

main_program{
  for(int i=0; i<5; i++){
  cout <<"Make offer: ";
  double p; cin >> p;
  if(p >= 500){
    cout <<"Accepted." <<endl;
    break;
  }
  double q = 1000 - p;
  cout <<"Will sell for "
       << q << endl;
  }
}
```

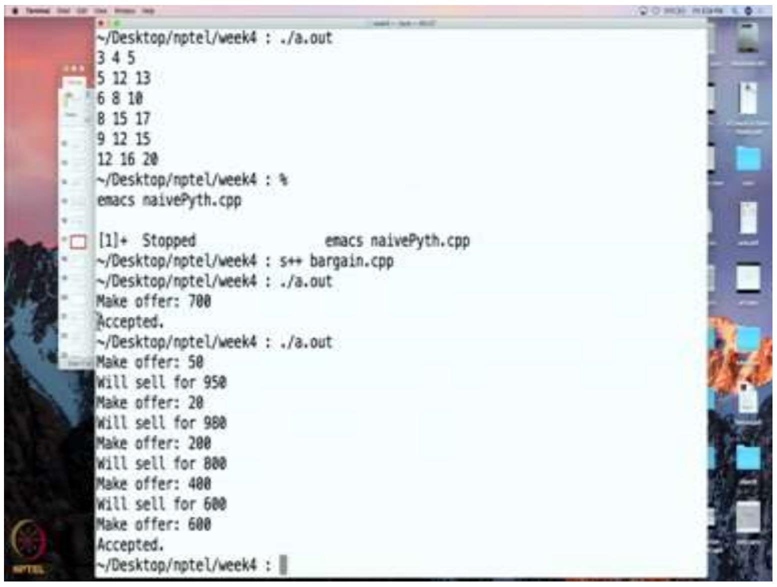
```
-UU-:----F1  bargain.cpp    All L1    (C++/l Abbrev) 3:22PM 1.34
```



```
5 12 13
6 8 10
8 6 10
8 15 17
9 12 15
12 5 13
12 9 15
12 16 20
15 8 17
16 12 20
~/Desktop/nptel/week4 : %
emacs naivePyth.cpp

[1]+  Stopped                 emacs naivePyth.cpp
~/Desktop/nptel/week4 : s++ pyth.cpp
~/Desktop/nptel/week4 : ./a.out
3 4 5
5 12 13
6 8 10
8 15 17
9 12 15
12 16 20
~/Desktop/nptel/week4 : %
emacs naivePyth.cpp

[1]+  Stopped                 emacs naivePyth.cpp
~/Desktop/nptel/week4 : s++ bargain.cpp
```



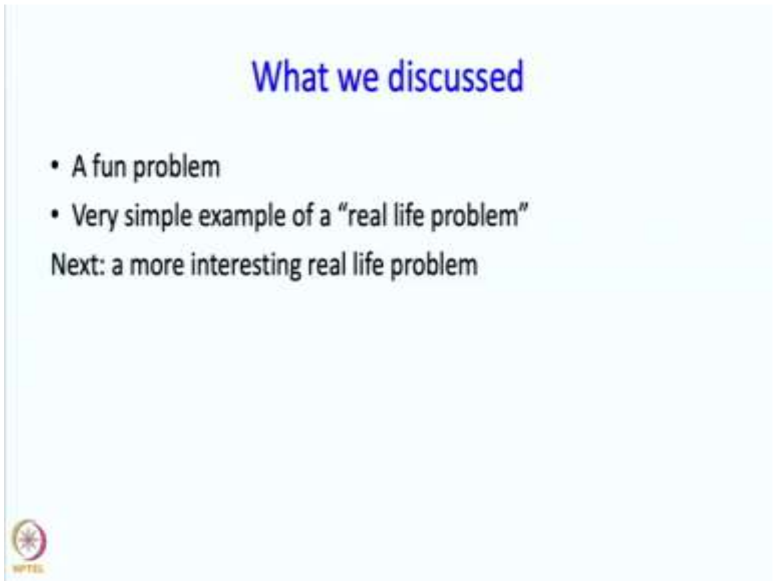
```
~/Desktop/nptel/week4 : ./a.out
3 4 5
5 12 13
6 8 10
8 15 17
9 12 15
12 16 20
~/Desktop/nptel/week4 : %
emacs naivePyth.cpp

[1]+  Stopped                 emacs naivePyth.cpp
~/Desktop/nptel/week4 : s++ bargain.cpp
~/Desktop/nptel/week4 : ./a.out
Make offer: 700
Accepted.
~/Desktop/nptel/week4 : ./a.out
Make offer: 50
Will sell for 950
Make offer: 20
Will sell for 980
Make offer: 200
Will sell for 800
Make offer: 400
Will sell for 600
Make offer: 600
Accepted.
~/Desktop/nptel/week4 :
```

So, let us see that, so this program I have coded as bargain.cpp and let see it, so this is the program so, let us compile it, and run it. So, it says make an offer, so what am I going to say? Maybe I will say 700, so already accepted. Because, why? Because we said that, the program accepts if you quote a high price or price 500 or more. So, let us try something different.

So this time let us be really annoying, let us say 50. If you are in the real market, and if you see something which is over 500, you know that this is a really low offer, but sometimes you do that, you are fooling around. So, at that time the seller also plays here, game and here, he says no no not 50, 950, what are you talking about? So, now you have to make another offer which is presumably more reasonable, so you might say 200, or sorry 20 I said, so of course sells a 980, but I mean 200. So, at that point a seller will say 800, so if you are reasonable then, the seller will also try to be a little bit more reasonable. May be at this point you say, 400. Then seller is even more reasonable. So, if you say 600 at this point seller will accept, if you had said something else, then as you know the loop would have go on, would have continued, and it would have stop after 5 iterations. Because, you cannot go on arguing at infinite term. Alright, so let us get back to the presentation.

(Refer Slide Time: 10:49)



What we discussed

- A fun problem
- Very simple example of a "real life problem"

Next: a more interesting real life problem

So, what have we discussed? So we discussed, a fun problem, it is a very simple example of a real life problem, it has this notion that, there is some system which is conversing with the user. So, the program is conversing with the user, just as the real life system is interacting with the user. But it is a really simple system, so that is okay, it is a little bit of fun. Next we are going to look at, a slightly more interesting real life problem. So, we will take quick break.