APPENDICES

Appendix-A

Derivation of expression for E(ti/T)

$E\left(\frac{t_i}{T}\right)$ can be computed as

$$E\left(\frac{t_i}{T}\right) = E\left(E\left(\frac{t_i}{T} \mid T = t\right)\right)$$

$$T = \sum_{j=1}^{n} t_j = t_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} t_j$$

t_i and $\sum_{\substack{j=1 \\ j \neq i}}^{n} t_j$ are independent random variables.

Where $t_i \sim Gamma(1, {}^{1}/_{\theta})$ and $\sum_{\substack{j=1 \\ j \neq i}}^{n} t_j \sim Gamma(n-1, {}^{1}/_{\theta})$

Let $x = u + v, y = v$

Joint distribution of u and v = Joint distribution of x and y.

$$conditional\ distribution\ of\ \ y/x\ = \frac{joint\ distribution\ of\ x\ \&\ y}{Marginal\ of\ x}$$

$v \sim Gamma\left(1, {}^{1}/_{\theta}\right)$

$f_y(v) = \frac{1}{\theta} e^{-v/\theta}$

$u \sim Gamma\left(n-1, {}^{1}/_{\theta}\right)$

$$g(u) = \frac{e^{-u/\theta}u^{n-2}}{\Gamma_{n-1}\theta^{n-1}}$$

$$f(u,v) = \frac{\frac{1}{\theta}e^{-v/\theta}\ e^{-u/\theta}u^{n-2}}{\Gamma_{n-1}\theta^{n-1}} = \frac{e^{-((u+v)/\theta)}u^{n-2}}{\Gamma_{n-1}\theta^{n}}$$

$$x = u + v\ \&\ y = v \Rightarrow v = x - y$$

$$f_{x,y}(x,y) = \frac{e^{-x/\theta}(x-y)^{n-2}}{\Gamma_{n-1}\theta^{n}}$$

$$conditional\ distribution\ of\ \ y/x\ = \frac{joint\ distribution\ of\ x\ and\ y}{Marginal\ of\ x}$$

$$= \frac{e^{-x/\theta}(x-y)^{n-2}}{\Gamma_{n-1}\theta^{n}} \frac{\Gamma_n\theta^n}{e^{-x/\theta}x^{n-1}}$$

$$= \frac{(x-y)^{n-2}(n-1)}{x^{n-1}}$$

$$= \frac{n-1}{x}\ (\frac{x-y}{x})^{n-2}$$

$$f_{y/x}(y) = \frac{n-1}{x}\ (1-\frac{y}{x})^{n-2} \quad ;\ \ 0 < y < x$$

Conditional distribution of $(t_i|T = t)$ is given by p.d.f

$$f_{(T_i|T)}(y) = \frac{n-1}{t}\left(1-\frac{y}{t}\right)^{n-2} \qquad ; 0 < y < t.$$

$$E(\mathrm{ti/T}) = \int_0^{\mathrm{t}} \frac{\mathrm{n}-1}{\mathrm{t}}\left(1-\frac{\mathrm{y}}{\mathrm{t}}\right)^{\mathrm{n}-2} \mathrm{y}dy$$

$$= n - 1 \int_0^t \left(1 - \frac{y}{t}\right)^{n-2} \frac{y}{t}\, dy$$

$$\frac{y}{t} = z \;\Rightarrow dy = tdz$$

$$E(\text{ti/T}) = \text{t(n} - 1) \int_0^1 (1 - \text{z})^{\overline{\text{n}-1}-1} \text{z dz}$$

$$= (n-1)t \frac{\Gamma_{n-1}\Gamma_n}{\Gamma_{n+,}}$$

$$= \frac{\Gamma_n\, t}{n\, \Gamma_n} \;=\; \frac{t}{n}$$

$$E\left(\frac{t_i}{T}\right) = E\left(E\left(\frac{t_i}{T}\middle|T\right)\right)$$

$$= E\left(\frac{1}{T}E(t_i|T)\right) = E\left(\frac{1}{t}\frac{t}{n}\right)$$

$$= E\left(\frac{1}{n}\right) = \frac{1}{n}$$

Appendix-B

C++ program to find optimum order quantity and total cost for the presented model in Chapter-2

```
#include<iostream>
using namespace std;
#include<cmath>
int main()
{
  double c0,ch,c,D;
  double n,tc,z;
  cout<<"Enter value of c0:";
  cin>>c0;
  cout<<"Enter value of ch:";
  cin>>ch;
  cout<<"Enter value of c:";
  cin>>c;
  cout<<"Enter value of D:" ;
  cin>>D;
  z= 2*D*c0/ch;
  Q=sqrt(z);
  cout<<"Q="<<Q;
  n= (1+sqrt(1+4*z))/2;
  cout<<" optimum Value n = "<<n<<endl; // :optimal
  order quantity.
  tc=c0*D/n+ D*c + (n+D)*(ch/2);
  cout<<"total cost= "<< tc;
  return 0;
 }
```

Appendix-C

Using this program we can find optimum inventory level and total cost model developed in Chapter-3

```
#include<iostream>
using namespace std;
#include<cmath>
int main()
{
  double c0,ch,c,theta;
  double n,etc,n1,n2,e1,e2;
  cout<<"Enter value of c0:";
  cin>>c0;
  cout<<"Enter value of ch:";
  cin>>ch;
  cout<<"Enter value of c:";
  cin>>c;
  cout<<"Enter value oftheta:" ;
  cin>>theta;
  n1=8*(c0+c)/(theta*ch);
  n2=sqrt(1+n1);
  n=(1+n2)/2;  //n : optimal order quantity.
  cout<<"value of n = "<<n<<endl;
  e1=(c0+n*c)/(theta*(n-1));
  e2=(n+1)*ch/2;
  etc=e1+e2;
  cout<<"Expected total cost = "<<etc;
  return 0;
 }
```

Appendix-D

Using this program we can find optimum inventory level and total cost model developed in Chapter-4

```
#include<iostream>
using namespace std;
#include<cmath>
int main()
{
  double c0,c1,c2,c,k,theta;
  double n1,etc,s1,s2,m1,m2,p,q,d1;
  int n;
  cout<<"Enter the value of c0 : ";
  cin>>c0;
  cout<<"Enter value of c1 : ";
   cin>>c1;
   cout <<"Enter value of c2 : ";
   cin>>c2;
   cout<<"Enter value of c : ";
   cin>>c;
   cout<<"Enter the value of theta : ";
   cin>>theta;
   cout<<"Enter the value of k :";
   cin>>k;
   p=(c1*((k*k)+k-1) - c2*k*(k+1)- 2*((c0+c)/theta))/c1;
   cout<< "value of p="<<p<<endl;
   q=(c2*k*(k+1) - c1*k*(k+1) - 2*(c0+c)/theta)/c1;
   cout<< "value of q="<<q<<endl;
   s1=sqrt(-p/3);
   cout<< "value of s1="<<s1<<endl;
   s2=sqrt(-3/p);
```

```
  cout<< "value of s2="<<s2<<endl;
  d1=3*q/(2*p);
  cout<< "value of d1="<<d1<<endl;
  m1= s2*d1;
  cout<< "value of m1="<<m1<<endl;
  m2=(1.0/3.0)*acos(m1);
  cout<<"m2 = "<<m2;
  n1 = 2*s1*cos(m2);
  cout<<"value of n1 ="<<n1;
  n = round(n1+0.5);
  cout<<" n="<<n;
  etc= (c0+(n*c))/(theta*(n-1)) + c1*(n-k)*(n+k+1)/(2*n)
        +c2*k*(k+1)/(2*n);
  cout<<"expected total cost ="<<etc<<endl;
  return 0;
}
```

Appendix-E

Using this program we can find optimum inventory level and total cost model developed in Chapter-5

```
#include<iostream>
using namespace std;
#include<cmath>
int main()
{
  double c0,c1,c2,c,t0, sum,term;
  int x,j;
  float theta,n,lm;
  double  e1,e2,e3,e4,e5,e6;
  double sum1, sum2;
  double etc[100];
  etc[0]= 15000000;
  cout<<"Enter the value of co : ";
  cin>>c0;
  cout<<"Enter value of c1 : ";
  cin>>c1;
  cout<<"Entert value of c2 :";
  cin>>c2;
  cout <<"Enter value of c : ";
  cin>>c;
  cout<<"Enter value of theta : ";
  cin>>theta;
  cout<<"Enter the value of t0 :";
  cin>>t0;
  lm=t0/theta;
  cout<<"value of lambda ="<<lm<<endl;
  n = 2;
  j=1;
  do
   {
    e1 = (c0 + n*c)/(theta*(n-1));
```

```
    e2 = n*lm*(c2-c1)/(n-1);
    e3 = c2*(n+1)/2 ;
    term = 1;
    sum =1;
    for(x=1; x<=n; x++)
    {
     term *= lm/x;
     sum += term; // compute kn
    }
    cout<<"sum = "<<sum;
    sum1=0;
    sum1= lm*(1-term/sum);
    e4= (c2-c1)*lm*sum1/(n-1);
    e5 = (c2-c1)* sum1/(2*n) ;
    sum2=sum1+(lm*lm)*(1-(term+term*n/lm)/sum);
    cout<<"sum1= "<<sum1 <<"  sum2 = "<<sum2;
    e6 = (c2-c1)*sum2/(2*n);
    etc[j] = e1 - e2 + e3 + e4 - e5 - e6;
    cout<<"n = :" << n <<"  "<< " E(tc): "<<etc[j]<<endl;
    cout<<"e1 = "<<e1 <<" e2 = "<<e2<<"e3 =  "<<e3<<endl;
    cout<<"e4= "<<e4<<"  e5 =  "<<e5<<" e6 = "<<e6<<endl;
    n++;
    j++;
  }
  while(etc[j-1]<etc[j-2]);
  n=n-1;
  cout<<"optimal value of n = "<<n-1<<endl;
  return 0 ;
}
```

Appendix-F

Using this program we can find optimum inventory level and total cost model developed in Chapter-6

```
#include<iostream>
using namespace std;
#include<cmath>
int main ()
{
  double c0,c1,c2,c,alpha,theta;
  int x,j;
  double p,term,n, e1,e2,e3,e4,e5;
  double sum1, sum2;
  double etc[100];
  etc[0]= 15000000;
  cout<<"Enter the value of c0 : ";
  cin>>c0;
  cout<<"Enter value of c1 : ";
  cin>>c1;
  cout <<"Enter value of c2 : ";
  cin>>c2;
  cout<<"Enter value of c : ";
  cin>>c;
  cout<<"Enter the value of theta : ";
  cin>>theta;
  cout<<"Enter the value of alpha :";
  cin>>alpha;
  n = 2;
  j=1;
  p=theta/(alpha+theta);
  do
   {
    e1 = (c0+(n*c))/(theta*(n-1));
    e2 = (c2-c1)*n*alpha/(theta*(n-1));
```

```
    e3 = c2*(n+1)/2.0;
    term = 1 ;
    sum1 = 0;
    for(x=1; x<=n; x++)
    {
     term *= (1-p);
     sum1 += x*term;
    }
    Sum1 *= p/ (1-term*(1-p));
    cout<< "term="<<term<<" sum1="<<sum1<<endl;
    e4 = (c2-c1)*alpha/(theta*(n-1))*sum1;
    term=1;
    sum2=0;
    for(x=1; x<=n; x++)
    {
     term *=(1-p);
     sum2+=(x*x*term);
     }
    sum2*= p/ (1-(term*(1-p)));
    cout<<"term = "<<term<<"  sum2 = "<<sum2<<endl;
    e5= (c2-c1)/(2*n)*(sum1+sum2);
    etc[j] = e1 - e2 + e3 + e4 - e5;
    cout<<e1<<"  "<<e2<<" "<<e3<<"  "<<e4<<" ”<<endl;
    cout<<"n = :" << n <<"  "<< " E(tc):”<< etc[j]<<endl;
    n++;
    j++;
   }
   while(etc[j-1]<etc[j-2]);
   n=n-1;
   cout<<"optimal value of n = "<<n-1<<endl;
   cout<<"value of etc ="<<etc[j-1];
   return 0 ;
}
```