INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD

DEPARTMENT OF SOFTWARE ENGINEERING

CS 211: Object Oriented Paradigm

Section: A

Submitted By: Hajira Gul (4454-FOC/BSSE-F22)
Areesha Abbasi (4508-FOC/BSSE-F22)

Submitted To: Ms. Shaista Rashid

Programming Project

Topic: THEME PARK MANAGEMENT SYSTEM

Topics Covered:

- Inheritance
- Polymorphism
- Operator Overloading
- File Reading and File
- Pure Virtual Function
- Exception Handling

Menu Options Included in our THEME PARK MANAGEMENT SYSTEM:

1. Display Information of All Attractions
2. Find Particular Attraction
3. Display Information of Cheapest Attraction
4. Display Information of Attraction with Maximum Seat Capacity
5. Display Information of Safe Attractions
6. Generate Timing Schedule of all Attractions
7. Average Price Per Attraction
8. Check Availability of Attraction in Theme Park
9. Purchase Ticket of an Attraction
10. Display Information of Attractions with Same Price
11. Increase or Decrease Price of an Attraction
12. Update Information
13. Write To New File

Code:

Attraction.h File:

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Attraction
{
   private:
   string Name;
   float Price_Per_Head;
   int Age;
   string Time_Duration;
   int Seat_Capacity;
   public:
   Attraction(string n,float p,int a,string t,int s)
   {
      Name=n;
      Price_Per_Head=p;
      Age=a;
      Time_Duration=t;
      Seat_Capacity=s;
   }
   virtual void GetInfo()
   {
      int error=0;
      do
      {
         try
         {
            cout<<"Enter Name of Attraction: ";
            getline(cin,Name);
            cout<<"Enter Time Duration: ";
            getline(cin,Time_Duration);
            cout<<"Enter Price of Attraction per head: ";
            cin>>Price_Per_Head;
            cout<<"Enter Age: ";
            cin>>Age;
            cout<<"Enter seat capacity of Attraction: ";
            cin>>Seat_Capacity;
            if (Price_Per_Head<0||Age<0||Seat_Capacity<0)
            {
               error = 1;
               throw error;
            }
            else
            {
               error=0;
            }
         }
         catch (int errorCode)
         {
```

```
         if(error==1)
         cout << "Error: Invalid input. Numeric values cannot be negative." << endl;
         else
         cout << "Unknown error." << endl;
         cin.ignore();

      }
   }
   while(error==1);
}
virtual void ShowInfo()
{
   cout<<"Name: "<<Name<<endl;
   cout<<"Price per head: "<<"Rs."<<Price_Per_Head<<endl;
   cout<<"Age: "<<Age<<"+"<<endl;
   cout<<"Time Duration: "<<Time_Duration<<endl;
   cout<<"Seat Capacity: "<<Seat_Capacity<<endl;
}
string TellName()
{
   return Name;
}
float TellPrice_Per_Head()
{
   return Price_Per_Head;
}
int TellAge()
{
   return Age;
}
string TellTime_Duration()
{
   return Time_Duration;
}
int TellSeat_Capacity()
{
   return Seat_Capacity;
}
int UpdateAndTellSeat_Capacity(int numTickets)
{
   Seat_Capacity -= numTickets;
   return Seat_Capacity;
}
virtual bool IsSafe()=0;
virtual string TypeOfAttraction()=0;
virtual float ChildAttribute()=0;
void operator+(float amount)
{
   Price_Per_Head=Price_Per_Head+amount;
}
void operator-(float amount)
{
   Price_Per_Head=Price_Per_Head-amount;
}
bool operator==( Attraction& a)
```

```
    {
        return (Price_Per_Head == a.Price_Per_Head);
    }
};
```

Ride.h File:

```
#include"Attraction.h"
#include<iostream>
using namespace std;
class Ride:public Attraction
{
    private:
    float Speed_Of_Ride;
    public:
    Ride(string n,float p,int a,string t,int s,float sor):Attraction(n,p,a,t,s)
    {
        Speed_Of_Ride=sor;
    }
    void GetInfo()
    {
        Attraction::GetInfo();
        cout<<"Enter Speed of ride: ";
        cin>>Speed_Of_Ride;
    }
    void ShowInfo()
    {
        Attraction::ShowInfo();
        cout<<"Speed of Ride: "<<Speed_Of_Ride<<"m/s"<<endl;
    }
    bool IsSafe()
    {
        if(Speed_Of_Ride<=100.0)
        return true;
        else
        return false;
    }
    string TypeOfAttraction()
    {
        return "Ride";
    }
    float ChildAttribute()
    {
        return Speed_Of_Ride;
    }
};
```

EntertainmentShow.h File:

```
#include"Attraction.h"
#include<iostream>
using namespace std;
class EntertainmentShow:public Attraction
{
    private:
    float Safety_Rating;
    public:
    EntertainmentShow(string n,float p,int a,string t,int s,float sr):Attraction(n,p,a,t,s)
```

```
   {
      Safety_Rating=sr;
   }
   void GetInfo()
   {
      Attraction::GetInfo();
      cout<<"Enter Safety Rating: ";
      cin>>Safety_Rating;
   }
   void ShowInfo()
   {
      Attraction::ShowInfo();
      cout<<"Safety Rating: "<<Safety_Rating<<"/10"<<endl;
   }
   bool IsSafe()
   {
      if(Safety_Rating>=5&&Safety_Rating<=10)
      return true;
      else
      return false;
   }
   string TypeOfAttraction()
   {
      return "Entertainment Show";
   }
   float ChildAttribute()
   {
      return Safety_Rating;
   }
};
```

ThemePark.h File:

```
#include"Attraction.h"
#include"Ride.h"
#include"EntertainmentShow.h"
#include<string>
#include<fstream>
using namespace std;
const int MAX=10;
class ThemePark
{
  Attraction *A[MAX];
   public:
   void ReadFromFile()
{
      string s1, s2, s3, s4, s5, s6, s7;
      ifstream ifile("Data.txt");
      for (int i = 0; i < MAX; i++)
{
         getline(ifile, s1, ',');
         getline(ifile, s2, ',');
         getline(ifile, s3, ',');
         getline(ifile, s4, ',');
         getline(ifile, s5, ',');
         getline(ifile, s6, ',');
         getline(ifile, s7);

         if (s1 == "Ride")
         {
            A[i] = new Ride(s2, stof(s3), stoi(s4), s5, stoi(s6), stof(s7));
         }
        else
        {
            A[i] = new EntertainmentShow(s2, stof(s3), stoi(s4), s5, stoi(s6), stof(s7));
         }
      }

      ifile.close();
}

   void DisplayAttractions()
   {
      for(int i=0;i<MAX;i++)
      {
         cout<<endl<<"*ATTRACTION: "<<i+1<<endl;
         A[i]->ShowInfo();
      }
   }
   void FindAttraction()
   {
      string a;
      bool found=false;
      cout<<"Enter The name of Attraction to be found: ";
      cin.ignore();
      getline(cin,a);
      for(int i=0;i<MAX&&found==false;i++)
      {
         if(A[i]->TellName()==a)
         {
            cout<<endl<<"*DESIRED ATTRACTION:"<<endl;
            A[i]->ShowInfo();
            found=true;
         }
      }
      if(found==false)
      cout<<"***No Such Attraction found !"<<endl;
   }
   void CheapAttraction()
   {
      float cheap=A[0]->TellPrice_Per_Head();
      int n=0;
      for(int i=0;i<MAX;i++)
      {
         if(cheap>A[i]->TellPrice_Per_Head())
         {
            cheap=A[i]->TellPrice_Per_Head();
            n=i;
         }
      }
      cout<<endl<<"*CHEAPEST ATTRACTION: "<<endl;
      A[n]->ShowInfo();
   }
   void MaxSeatattraction()
   {
      int max=A[0]->TellSeat_Capacity();
      int k=0;
      for(int i=0;i<MAX;i++)
      {
         if(max<A[i]->TellSeat_Capacity())
         {
            max=A[i]->TellSeat_Capacity();
            k=i;
         }
      }
      cout<<endl<<"*ATTRACTION WITH MAXIMUM SEAT CAPACITY: "<<endl;
      A[k]->ShowInfo();
      cout<<endl<<"Updated Seat Capacity: "<<A[k]-
>UpdateAndTellSeat_Capacity(0)<<endl;
   }
   void SafeAttraction()
   {
      bool found=false;
      cout<<endl<<"*SAFE ATTRACTIONS: "<<endl;
      for(int i=0;i<MAX;i++)
      {
         if(A[i]->IsSafe()==true)
         {

            cout<<endl<<"ATTRACTION: "<<i+1<<endl;
            A[i]->ShowInfo();
            found=true;
         }
      }
      if(found==false)
      cout<<"No Safe Attractions found"<<endl;
   }
   void UpdateInfo()
   {
      string h;
      bool found=false;
      cout<<"Enter name of Attraction whose information you want to update: ";
      cin.ignore();
      getline(cin,h);
      for(int i=0;i<MAX&&found==false;i++)
      {
         if(h==A[i]->TellName())
         {
            cout<<"*CURRENT INFORMATION: "<<endl;
            A[i]->ShowInfo();
            cout<<" "<<endl;
            cout<<"Enter the updated Information:"<<endl;
            A[i]->GetInfo();
            found=true;
            cout<<endl<<"***NOW THE UPDATED INFORMATION IS AS FOLLOWS:
```

```
"<<endl;
        A[i]->ShowInfo();
        WriteInExistingFile();
      }
    }
    if(found==false)
    cout<<"No Attraction with this name found in Theme Park"<<endl;
  }
  void WriteInExistingFile()
  {
    ofstream ofile("Data.txt");
    for (int i = 0; i < MAX; i++)
    {
      ofile << A[i]->TypeOfAttraction() << "," << A[i]->TellName() << ","
       << A[i]->TellPrice_Per_Head() << "," << A[i]->TellAge() << ","
       << A[i]->TellTime_Duration() << "," << A[i]->TellSeat_Capacity() << ",";

    if (A[i]->TypeOfAttraction() == "Ride")
    {
      ofile<<A[i]->ChildAttribute()<<endl;
    }
    else if (A[i]->TypeOfAttraction() == "Entertainment Show")
    {
      ofile<<A[i]->ChildAttribute()<<endl;
    }


  }
  ofile.close();
  }
  void TimingSchedule()
  {
    cout<<endl<<"----------------------------------------"<<endl;
     cout<<"*ATTRACTION*"<<"\t"<<"\t"<<"*TIMING SCHEDULE*"<<endl;

    for(int i=0;i<MAX;i++)
    {
      cout<<"|";
      cout<<A[i]->TellName()<<"\t"<<"\t"<<A[i]->TellTime_Duration();
      cout<<"|"<<endl;
    }
    cout<<"--------------------------------------"<<endl;
  }
  void Availability()
  {
    string AttractionName;
    bool found=false;
    cout<<endl<<"Enter Name of Attraction to check its Availability: ";
    cin.ignore();
    getline(cin,AttractionName);
    for(int i=0;i<MAX;i++)
    {
      if(AttractionName==A[i]->TellName())
      {
        cout<<endl<<"*ATTRACTION IS AVAILABLE IN THE THEME PARK WITH
```

```
FOLLOWING INFORMATION:"<<endl;
        A[i]->ShowInfo();
        if(A[i]->TellSeat_Capacity()==0)
        {
          cout<<endl<<"*THIS ATTRACTION IS FULLY OCCUPIED!"<<endl;
        }
        else
        {
          cout<<endl<<"*YES,THIS ATTRACTION IS AVAILABLE!"<<endl;
          cout<<"Attraction is Available with Seat Capacity of "<<A[i]-
>TellSeat_Capacity()<<endl;
        }
        cout<<endl;
        found=true;
      }
    }
    if(found==false)
    cout<<"No Attraction with name found in Theme Park"<<endl;
  }
  void PurchaseTicket()
  {
    string AttractionName;
    int numTickets;
    bool found=false;
    cout<<endl<<"Enter Name of Attraction to check its Availability: ";
    cin.ignore();
    getline(cin,AttractionName);
    for(int i=0;i<MAX;i++)
    {
      if(AttractionName==A[i]->TellName())
      {
        cout<<endl<<"*ATTRACTION IS AVAILABLE IN THE THEME PARK WITH
FOLLOWING INFORMATION:"<<endl;
        A[i]->ShowInfo();
        if(A[i]->TellSeat_Capacity()>0)
        {
          cout<<endl<<"--How Many Tickets you want to purchase: ";
          cin>>numTickets;
          if (numTickets<= A[i]->TellSeat_Capacity())
          {
           cout<<"________________________________________"<<endl;
           cout<<endl<<"***ATTRACTION TICKET***"<<endl;
           cout<<"Number of Ticket Purchased: "<<numTickets<<endl;
           cout<<"Attraction: "<<AttractionName<<endl;
           cout<<"Price of One Ticket"<<A[i]->TellPrice_Per_Head()<<endl;
           cout<<endl<<"*TOTAL AMOUNT OF TICKET: "<<A[i]-
>TellPrice_Per_Head()*numTickets<<endl;
           cout<<"***TICKET PURCHASED SUCCESSFULLY!***"<<endl;
           cout<<"***ENJOY YOUR VISIT!***"<<endl;
           cout<<"________________________________________"<<endl;
           int updatedSeatCapacity = A[i]->UpdateAndTellSeat_Capacity(numTickets);
           cout << endl << "Now the remaining Seats available: " <<
updatedSeatCapacity << endl;
           WriteInExistingFile();
          }
```

```
               else
               {
                  cout << endl << "Sorry, not enough seats available for the requested number
of tickets." << endl;
               }
            }

            else
            {
               cout<<endl<<"*THIS ATTRACTION IS FULLY OCCUPIED!.....NO SEAT
AVAILABLE!"<<endl;
            }
            cout<<endl;
            found=true;
         }
      }
      if(found==false)
      cout<<"No Attraction with name found in Theme Park"<<endl;

   }
    void WriteToFile()
   {
      ofstream ofile("ThemeParkAttractions.txt");
      for(int i=0;i<MAX;i++)
      {
         ofile<<endl<<"ATTRACTION:"<<i+1<<endl;
         ofile<<"-"<<A[i]->TypeOfAttraction()<<":"<<endl;
         ofile<<"--"<<A[i]->TellName()<<endl;
         ofile<<"Price per head: "<<"Rs."<<A[i]->TellPrice_Per_Head()<<endl;
         ofile<<"Age: "<<A[i]->TellAge()<<endl;
         ofile<<"Time Duration: "<<A[i]->TellTime_Duration()<<endl;
         ofile<<"Seats Capacity: "<<A[i]->TellSeat_Capacity()<<endl;
         if(A[i]->TypeOfAttraction()=="Ride")
         {
            ofile<<"Speed Of Ride: "<<A[i]->ChildAttribute()<<"m/s"<<endl;
         }
         else if(A[i]->TypeOfAttraction()=="Entertainment Show")
         {
            ofile<<"Safety Rating: "<<A[i]->ChildAttribute()<<"/10"<<endl;
         }
      }
      cout<<"Information Successfully Added!"<<endl;
      ofile.close();
   }
   void UpdatePrice()
   {
      string attractionName;
      float amount;
      char z;
      cout<<"Enter the name of the attraction: ";
      cin.ignore();
      getline(cin, attractionName);
      bool found = false;
      for(int i=0;i<MAX&&found==false;i++)
      {
         if (attractionName==A[i]->TellName())
         {
            cout<<endl<<"*Enter a for Addition and s For Subtraction*"<<endl;
            cout<<"Enter choice to increase and decrease Price (a or s): ";
            cin>>z;
            if(z=='a')
            {
               cout<<"Enter the amount to increase the price: Rs.";
               cin>>amount;
               A[i]->operator+(amount);
               cout<<"PRICE INCREASED."<<endl;
               cout<<endl<<"Updated information:"<<endl;
               A[i]->ShowInfo();
               WriteInExistingFile();
            }
            else if(z=='s')
            {
               cout<<"Enter the amount to decrease the price: Rs.";
               cin>>amount;
               if(amount<=A[i]->TellPrice_Per_Head())
            {
               A[i]->operator-(amount);
               cout<<"PRICE DECREASED."<<endl;
               cout<<endl<<"Updated information:"<<endl;
               A[i]->ShowInfo();
               WriteInExistingFile();
            }
            else
            {
               cout<<"*Invalid Decrement in the Price"<<endl;
            }
            }
            else
            {
               cout<<"Invalid Command"<<endl;
            }
            found = true;
         }
      }

      if (found==false)
      {
         cout<<"Attraction not found."<<endl;
      }
   }

  void DisplayAttractionsWithSamePrice()
 {
    int z;
    bool found = false;
    for(int i=0;i<MAX;i++)
    {
      for(int j=i+1;j<MAX;j++)
      {
         if (A[i]->TellPrice_Per_Head()==A[j]->TellPrice_Per_Head())
         {
            z++;
            cout<<endl<<"Attractions with the same price:"<<endl;
            cout<<"----------------------------------------"<<endl;
            cout<<endl<<"*ATTRACTION 01: "<<endl;
            A[i]->ShowInfo();
            cout<<endl<<"*ATTRACTION 02: "<<endl;
            A[j]->ShowInfo();
            cout<<"----------------------------------------"<<endl;
            found=true;

         }

      }
   }
      cout<<endl<<"=> Set Of Events Occurring On Same Day: "<<z<<endl;
 }
void ViewStatistics()
{
   float totalPrice = 0;
   int Count = 0;

   for (int i = 0; i < MAX; i++)
   {
      totalPrice=totalPrice+A[i]->TellPrice_Per_Head();
      Count++;
   }

   if (Count>0)
   {
      float averagePrice=totalPrice/Count;

      cout<<endl;
      cout<<"----------------------------------------"<<endl;
      cout<<"->Average Price per Attraction: Rs." <<averagePrice<<endl;
      cout<<"----------------------------------------"<<endl;
   }
   else
   {
      cout<<"No attractions to calculate statistics."<<endl;
   }
}
~ThemePark()
 {
      for(int i=0;i<MAX;i++)
      delete A[i];
 }
};
```

Main.cpp File:

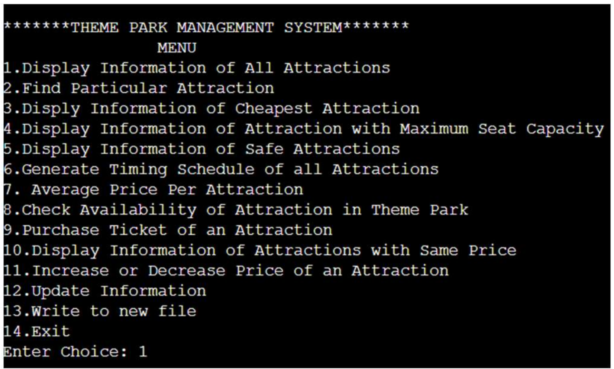
```
#include"ThemePark.h"
#include<iostream>
using namespace std;
int main()
```

```
{
   ThemePark T;
   int choice;
   T.ReadFromFile();
   do
   {
      cout<<endl<<"*******THEME PARK MANAGEMENT SYSTEM*******"<<endl;
      cout<<"\t"<<"\t"<<"MENU"<<endl;
      cout<<"1.Display Information of All Attractions"<<endl;
      cout<<"2.Find Particular Attraction "<<endl;
      cout<<"3.Disply Information of Cheapest Attraction"<<endl;
      cout<<"4.Display Information of Attraction with Maximum Seat Capacity"<<endl;
      cout<<"5.Display Information of Safe Attractions"<<endl;
      cout<<"6.Generate Timing Schedule of all Attractions"<<endl;
      cout<<"7. Average Price Per Attraction"<<endl;
      cout<<"8.Check Availability of Attraction in Theme Park"<<endl;
      cout<<"9.Purchase Ticket of an Attraction"<<endl;
      cout<<"10.Display Information of Attractions with Same Price"<<endl;
      cout<<"11.Increase or Decrease Price of an Attraction"<<endl;
      cout<<"12.Update Information"<<endl;
      cout<<"13.Write to new file"<<endl;
      cout<<"14.Exit"<<endl;
      cout<<"Enter Choice: ";
      cin>>choice;
      if(choice==1)
      T.DisplayAttractions();
      else if(choice==2)
      T.FindAttraction();
      else if(choice==3)
      T.CheapAttraction();
      else if(choice==4)
      T.MaxSeatattraction();
      else if(choice==5)
      T.SafeAttraction();
      else if(choice==6)
      T.TimingSchedule();
      else if(choice==7)
      T.ViewStatistics();
      else if(choice==8)
      T.Availability();
      else if(choice==9)
      T.PurchaseTicket();
      else if(choice==10)
      T.DisplayAttractionsWithSamePrice();
      else if(choice==11)
      T.UpdatePrice();
      else if (choice==12)
      T.UpdateInfo();
      else if(choice==13)
      T.WriteToFile();
   }
   while(choice!=14);
   cout<<endl<<"*******THANK YOU FOR USING THEME PARK MANAGEMENT
SYSTEM******"<<endl;
}
```

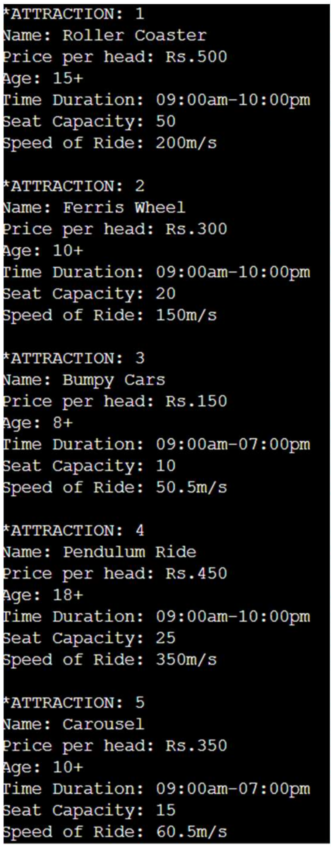
Data.txt:

```
Ride,Roller Coaster,500,15,09:00am-10:00pm,50,200
Ride,Ferris Wheel,300,10,09:00am-10:00pm,20,150
Ride,Bumpy Cars,150,8,09:00am-07:00pm,10,50.5
Ride,Pendulum Ride,450,18,09:00am-10:00pm,25,350
Ride,Carousel,350,10,09:00am-07:00pm,15,60.5
Entertainment Show,Haunted House,1000,12,04:00pm-12:00am,60,3
Entertainment Show,Fun House,500,3,09:00am-10:00pm,50,8
Entertainment Show,Magic Show,450,8,02:00pm-09:00pm,20,6
Entertainment Show,Fountain Show,600,10,07:00pm-12:00am,40,4
Entertainment Show,Puppet Show,700,4,12:00pm-12:00am,35,7
```

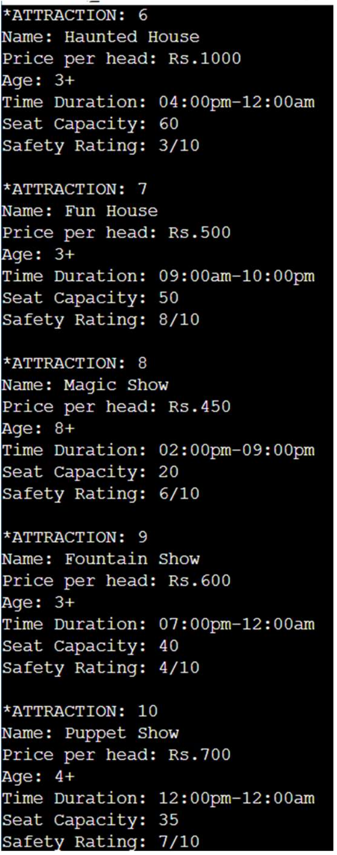
OUTPUT:



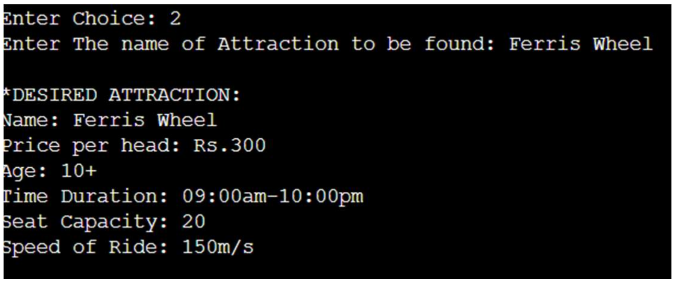
```
*******THEME PARK MANAGEMENT SYSTEM*******
                MENU
1.Display Information of All Attractions
2.Find Particular Attraction
3.Disply Information of Cheapest Attraction
4.Display Information of Attraction with Maximum Seat Capacity
5.Display Information of Safe Attractions
6.Generate Timing Schedule of all Attractions
7. Average Price Per Attraction
8.Check Availability of Attraction in Theme Park
9.Purchase Ticket of an Attraction
10.Display Information of Attractions with Same Price
11.Increase or Decrease Price of an Attraction
12.Update Information
13.Write to new file
14.Exit
Enter Choice: 1
```



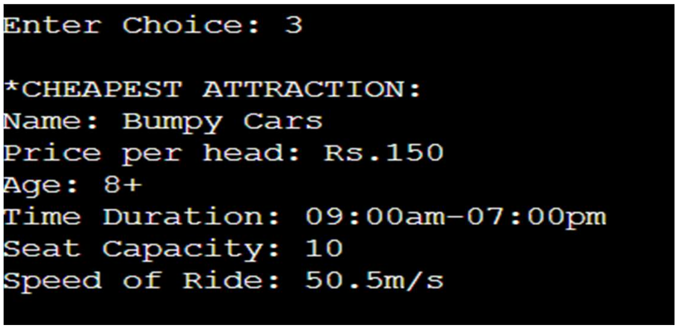
```
*ATTRACTION: 1
Name: Roller Coaster
Price per head: Rs.500
Age: 15+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Speed of Ride: 200m/s

*ATTRACTION: 2
Name: Ferris Wheel
Price per head: Rs.300
Age: 10+
Time Duration: 09:00am-10:00pm
Seat Capacity: 20
Speed of Ride: 150m/s

*ATTRACTION: 3
Name: Bumpy Cars
Price per head: Rs.150
Age: 8+
Time Duration: 09:00am-07:00pm
Seat Capacity: 10
Speed of Ride: 50.5m/s

*ATTRACTION: 4
Name: Pendulum Ride
Price per head: Rs.450
Age: 18+
Time Duration: 09:00am-10:00pm
Seat Capacity: 25
Speed of Ride: 350m/s

*ATTRACTION: 5
Name: Carousel
Price per head: Rs.350
Age: 10+
Time Duration: 09:00am-07:00pm
Seat Capacity: 15
Speed of Ride: 60.5m/s
```



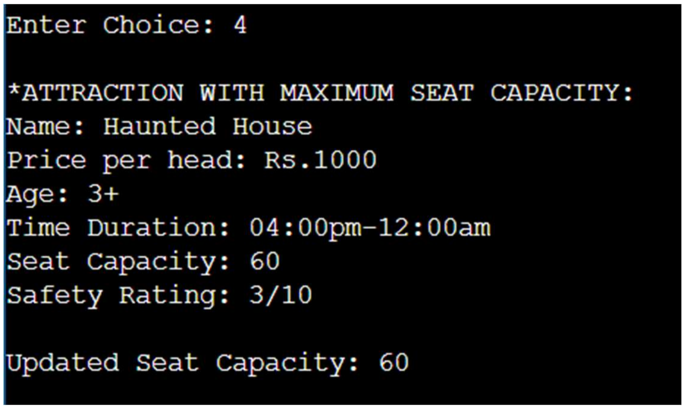
```
*ATTRACTION: 6
Name: Haunted House
Price per head: Rs.1000
Age: 3+
Time Duration: 04:00pm-12:00am
Seat Capacity: 60
Safety Rating: 3/10

*ATTRACTION: 7
Name: Fun House
Price per head: Rs.500
Age: 3+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Safety Rating: 8/10

*ATTRACTION: 8
Name: Magic Show
Price per head: Rs.450
Age: 8+
Time Duration: 02:00pm-09:00pm
Seat Capacity: 20
Safety Rating: 6/10

*ATTRACTION: 9
Name: Fountain Show
Price per head: Rs.600
Age: 3+
Time Duration: 07:00pm-12:00am
Seat Capacity: 40
Safety Rating: 4/10

*ATTRACTION: 10
Name: Puppet Show
Price per head: Rs.700
Age: 4+
Time Duration: 12:00pm-12:00am
Seat Capacity: 35
Safety Rating: 7/10
```



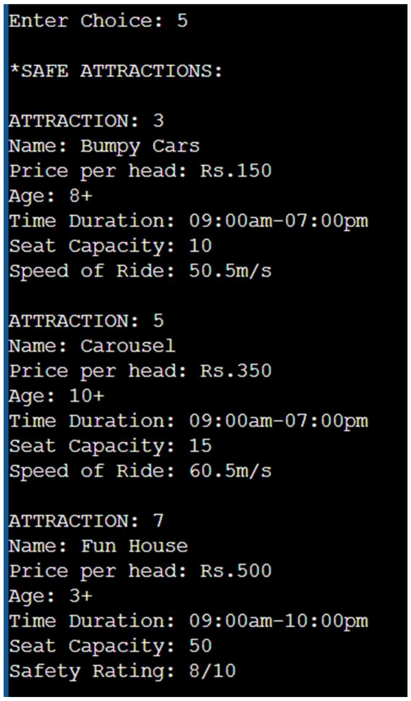
```
Enter Choice: 2
Enter The name of Attraction to be found: Ferris Wheel

*DESIRED ATTRACTION:
Name: Ferris Wheel
Price per head: Rs.300
Age: 10+
Time Duration: 09:00am-10:00pm
Seat Capacity: 20
Speed of Ride: 150m/s
```



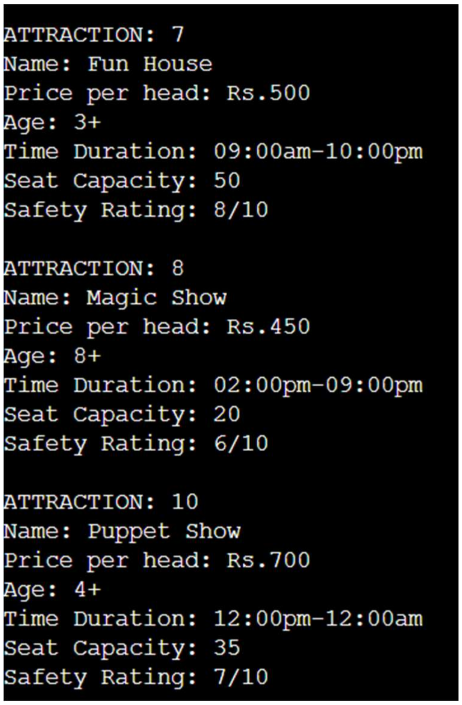
```
Enter Choice: 3

*CHEAPEST ATTRACTION:
Name: Bumpy Cars
Price per head: Rs.150
Age: 8+
Time Duration: 09:00am-07:00pm
Seat Capacity: 10
Speed of Ride: 50.5m/s
```



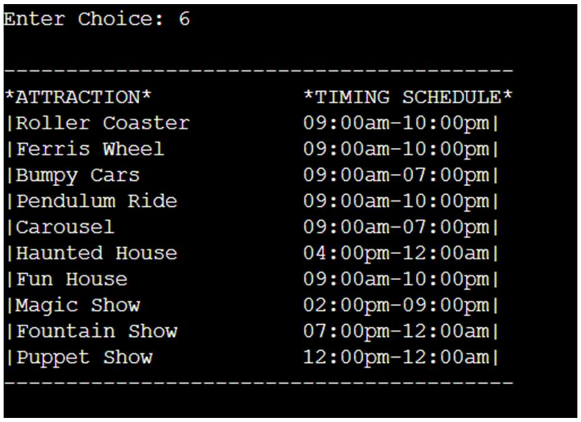
```
Enter Choice: 4

*ATTRACTION WITH MAXIMUM SEAT CAPACITY:
Name: Haunted House
Price per head: Rs.1000
Age: 3+
Time Duration: 04:00pm-12:00am
Seat Capacity: 60
Safety Rating: 3/10

Updated Seat Capacity: 60
```



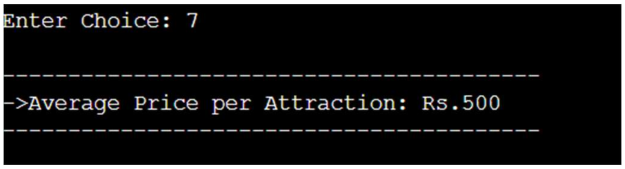
```
Enter Choice: 5

*SAFE ATTRACTIONS:

ATTRACTION: 3
Name: Bumpy Cars
Price per head: Rs.150
Age: 8+
Time Duration: 09:00am-07:00pm
Seat Capacity: 10
Speed of Ride: 50.5m/s

ATTRACTION: 5
Name: Carousel
Price per head: Rs.350
Age: 10+
Time Duration: 09:00am-07:00pm
Seat Capacity: 15
Speed of Ride: 60.5m/s

ATTRACTION: 7
Name: Fun House
Price per head: Rs.500
Age: 3+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Safety Rating: 8/10
```



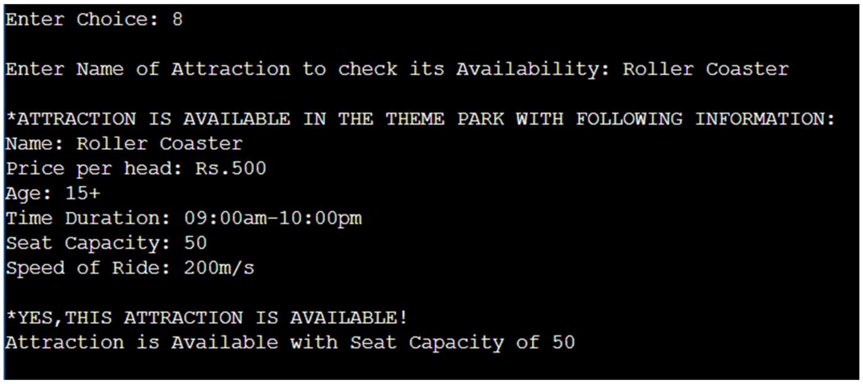
```
ATTRACTION: 7
Name: Fun House
Price per head: Rs.500
Age: 3+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Safety Rating: 8/10

ATTRACTION: 8
Name: Magic Show
Price per head: Rs.450
Age: 8+
Time Duration: 02:00pm-09:00pm
Seat Capacity: 20
Safety Rating: 6/10

ATTRACTION: 10
Name: Puppet Show
Price per head: Rs.700
Age: 4+
Time Duration: 12:00pm-12:00am
Seat Capacity: 35
Safety Rating: 7/10
```



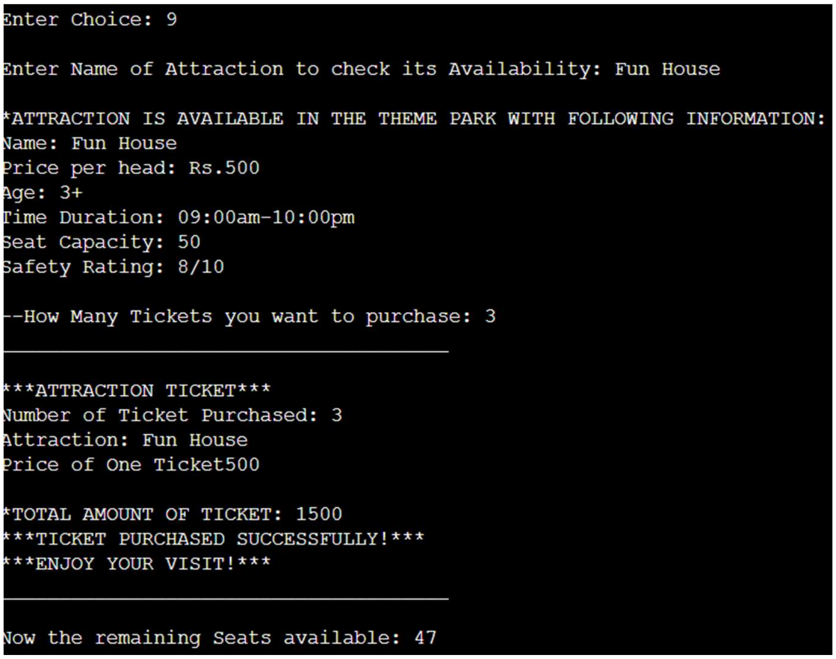
```
Enter Choice: 6

-----------------------------------------
*ATTRACTION*            *TIMING SCHEDULE*
|Roller Coaster         09:00am-10:00pm|
|Ferris Wheel           09:00am-10:00pm|
|Bumpy Cars             09:00am-07:00pm|
|Pendulum Ride          09:00am-10:00pm|
|Carousel               09:00am-07:00pm|
|Haunted House          04:00pm-12:00am|
|Fun House              09:00am-10:00pm|
|Magic Show             02:00pm-09:00pm|
|Fountain Show          07:00pm-12:00am|
|Puppet Show            12:00pm-12:00am|
-----------------------------------------
```



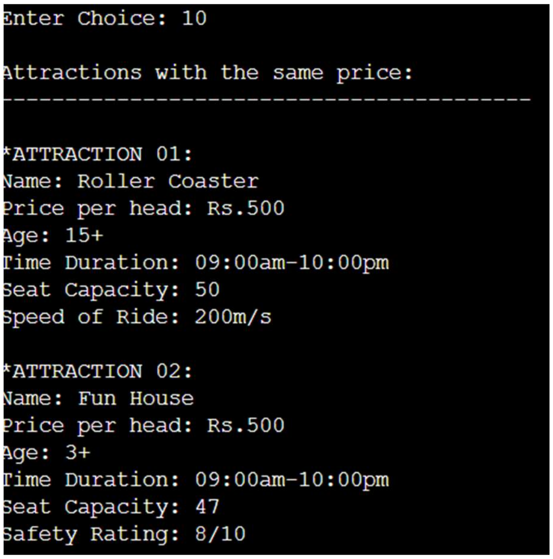
```
Enter Choice: 7

-----------------------------------------
->Average Price per Attraction: Rs.500
-----------------------------------------
```



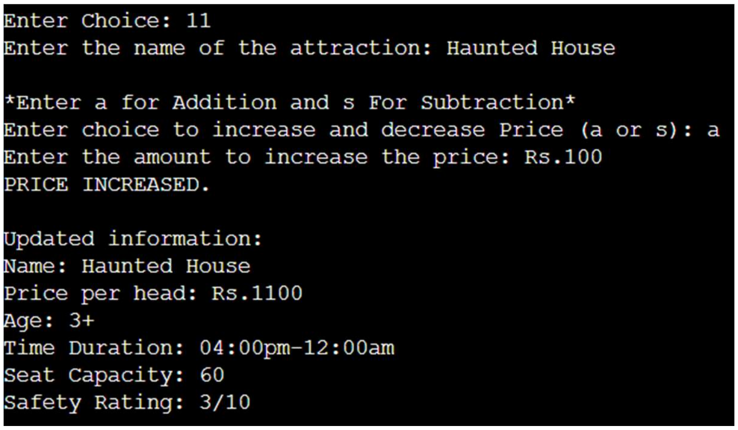
```
Enter Choice: 8

Enter Name of Attraction to check its Availability: Roller Coaster

*ATTRACTION IS AVAILABLE IN THE THEME PARK WITH FOLLOWING INFORMATION:
Name: Roller Coaster
Price per head: Rs.500
Age: 15+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Speed of Ride: 200m/s

*YES,THIS ATTRACTION IS AVAILABLE!
Attraction is Available with Seat Capacity of 50
```



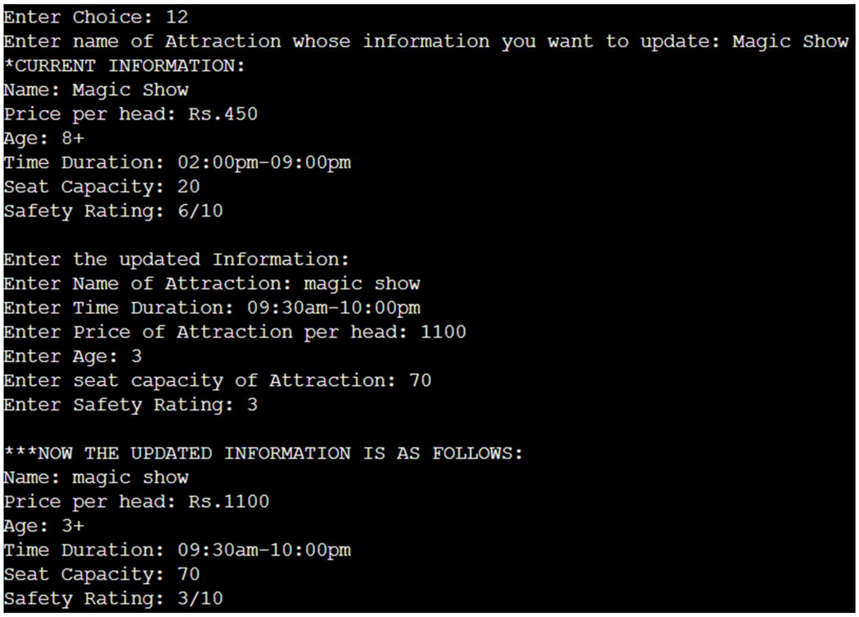
```
Enter Choice: 9

Enter Name of Attraction to check its Availability: Fun House

*ATTRACTION IS AVAILABLE IN THE THEME PARK WITH FOLLOWING INFORMATION:
Name: Fun House
Price per head: Rs.500
Age: 3+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Safety Rating: 8/10

--How Many Tickets you want to purchase: 3
______________________________________

***ATTRACTION TICKET***
Number of Ticket Purchased: 3
Attraction: Fun House
Price of One Ticket500

*TOTAL AMOUNT OF TICKET: 1500
***TICKET PURCHASED SUCCESSFULLY!***
***ENJOY YOUR VISIT!***
______________________________________

Now the remaining Seats available: 47
```



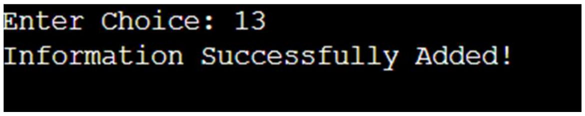
```
Enter Choice: 10

Attractions with the same price:
-----------------------------------------

*ATTRACTION 01:
Name: Roller Coaster
Price per head: Rs.500
Age: 15+
Time Duration: 09:00am-10:00pm
Seat Capacity: 50
Speed of Ride: 200m/s

*ATTRACTION 02:
Name: Fun House
Price per head: Rs.500
Age: 3+
Time Duration: 09:00am-10:00pm
Seat Capacity: 47
Safety Rating: 8/10
```



```
Enter Choice: 11
Enter the name of the attraction: Haunted House

*Enter a for Addition and s For Subtraction*
Enter choice to increase and decrease Price (a or s): a
Enter the amount to increase the price: Rs.100
PRICE INCREASED.

Updated information:
Name: Haunted House
Price per head: Rs.1100
Age: 3+
Time Duration: 04:00pm-12:00am
Seat Capacity: 60
Safety Rating: 3/10
```



```
Enter Choice: 12
Enter name of Attraction whose information you want to update: Magic Show
*CURRENT INFORMATION:
Name: Magic Show
Price per head: Rs.450
Age: 8+
Time Duration: 02:00pm-09:00pm
Seat Capacity: 20
Safety Rating: 6/10

Enter the updated Information:
Enter Name of Attraction: magic show
Enter Time Duration: 09:30am-10:00pm
Enter Price of Attraction per head: 1100
Enter Age: 3
Enter seat capacity of Attraction: 70
Enter Safety Rating: 3

***NOW THE UPDATED INFORMATION IS AS FOLLOWS:
Name: magic show
Price per head: Rs.1100
Age: 3+
Time Duration: 09:30am-10:00pm
Seat Capacity: 70
Safety Rating: 3/10
```



```
Enter Choice: 13
Information Successfully Added!
```