

THEME PARK MANAGEMENT

PROJECT USING C++

DONE BY:
VIVEKA P

INTRODUCTION

THEME PARK MANAGEMENT

This project on theme park management system has been developed on C++ with classes and Binary file handling techniques. The main objective of developing this project is to manage the details visitors of the theme park. The purpose of this project is to build an application program to reduce the manual work for managing details through programming. This application has a good appearance and is very easy to operate. It presents no difficulty and is easy to access in any platform. It is a very simple source code. This project provides a lot of features to manage it in a well manner. Every possible care has been taken to make the software and the report clear, simple and error free which makes it easy to understand and operate

OBJECTIVES OF THE PROJECT

To help the theme park management in making their business cost effective and easy to operate.

- The proposed system should maintain all records and transactions and should generate the required reports and information when required.
- It will also show the attitude of the management that they are aware of the newly introduced technology and ready to adopt them.
- To keep a record of the total number of customers who buy our theme park for every financial year.

PURPOSE OF THE PROJECT

- Electronically handling of theme park records, to enhance the accuracy, flexibility, reliability and to remove the human errors.
- To provide accurate information about the addition, deletion and modification of theme park records.
- To provide efficient, accurate, reliable, fast and robust structure that can handle any number of transactions

SYSTEM REQUIREMENTS

HARDWARE USED:

While developing the system, the used hardware is: PC with Core-i3-3120M CPU processor having 4.00 GB RAM and other required devices

SOFTWARE USED:

- Microsoft windows 7 as operating system.
- Turbo C++ , Version 4.0 running on DOSBOX in Windows 7.
- MS Word 2007 - for documentation

NEED FOR COMPUTERISATION

A few factors that direct us to develop a new system are given below:

- Faster system
- Accuracy
- Reliability
- Informative
- Reservations and cancellations from anywhere to anyplace

PROJECT OVERVIEW

- I. Databases and database systems have become an essential component of everyday life in modern society. We nowadays deal with activities which involve more use of database. Some real life examples are theme park and railway reservations, movie booking, etc. where we use databases.
- II. The above interactions are examples that we call as traditional database applications that is stored and accessed is either textual or numeric. In our project we concentrate more on computer application.
- III. There are several ways to implement databases. Some of them are file handling mechanism. Relational database or object oriented databases. Our project deals with file handling feature provided by C++.
- IV. The entire project on theme park management is shown through the medium of this project. It provides the facilities to add, modify, delete and search details at the theme park management.

SYSTEM DESIGN AND DEVELOPMENT

HEADER FILES USED AND THEIR PURPOSE

1. **FSTREAM.H** – For file handling, cin and cout
2. **PROCESS.H** – For exit() function
3. **CONIO.H** – For clrscr() and getch() functions
4. **STDIO.H** – For standard I/O operations
5. **STRING.H** For string handling
6. **CTYPE.H** – For character handling

Class Design:

class information

{

public:

int pn;

char n[100];

char a[150];

int ta;

```

void getinfo()
{
cout<<"Enter name\n";
gets(n);
cout<<"Enter address\n";
gets(a);
cout<<"Enter phone number\n";
cin>>pn;
}

void amount(int a1,int a2,int a3,int a4,int a5)
{
ta=a1+a2+a3+a4+a5;
}

void showdetails()
{
cout<<pn<<endl<<a<<endl<<n<<endl<<ta;
}};

```

Table Design- The databases of theme park management system consists of some tables. The table is designed to

store room detail records. The table and its structure is given below-

Files Used:

- **Parks.dat**

Fields in the file:

Operations Performed on Files:

- **Writing**
- **Reading**

USER MANUAL

WORKING WITH SOFTWARE PROJECT:

The theme park management program consists of the following logically organised menu-structure for the easy functionality. User may chose the menu options for the corresponding works.

Main menu- It directs you to supervise the theme park by showing current details of all the options in terms of shops available in the park.

New visitors- It helps in adding a new visitor to the theme park details.

Details of all Theme parks- It helps you to check all details of the visitors and shops currently available.

Specific details of Theme park- It helps to display the details of the specific visitor by the use of the identification terms (E.g.: Name, Phone no., Address, etc).

Add to Existing Detail- It helps in adding information to an already existing member in the theme park.

SOURCE CODE :

```
#include<fstream.h>
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
#include<stdio.h>
#include<string.h>
#include<process.h>
#include<iomanip.h>
int a,x1,x2,x3,x4,x5;
class information
{
public:
int pn;
char n[100];
char a[150];
int ta;
char date[10];
void getinfo()
{
cout<<"Enter name\n";
gets(n);
cout<<"Enter address\n";
```

```

gets(a);
cout<<"Enter phone number\n";
cin>>pn;
cout<<"Enter Date\n";
gets(date);
}

void amount(int a1,int a2,int a3,int a4,int a5)
{
ta=a1+a2+a3+a4+a5;
}

void showdetails()
{
cout<<"Phone Number:"<<pn<<endl;
cout<<" Name :"<<n<<endl;
cout<<"Address :"<<a<<endl;
cout<<"Date   :"<<date<<endl;
cout<<"Total Amount collected:"<<ta<<endl;
}

```

```
}i;  
class souvenirs  
{  
private:  
int n1,n2,n3,n4,n5,n6,c1,c2,c3,c4,c5,c6,x,total;  
public:  
souvenirs()  
{  
n1=0;  
n2=0;  
n3=0;  
n4=0;  
n5=0;  
n6=0;  
c1=0;  
c2=0;  
c3=0;  
c4=0;
```

```

c5=0;
c6=0;
}

// Where n = number of each item and c = cost of
item //f

int amount1()
{
return total;
}

void getdata();
void delete1();
void bill();
}s;

void souvenirs::getdata()
{
cout<<"\t"<<"Welcome to SMVP
Park"<<"\t"<<endl;

cout<<"\t"<<"SMVP Shops"<<"\t"<<endl;

```

```
cout<<"Enter the products:"<<endl;
cout<<"1.SMVP Hat"<<endl;
cout<<"2.SMVP Mug"<<endl;
cout<<"3.SMVP Large bag"<<endl;
cout<<"4.SMVP Photo"<<endl;
cout<<"5.SMVP Photo Frame"<<endl;
cout<<"6.SMVP Bracelet"<<endl;
cout<<"7.Exit"<<endl;
do
{
cin>>x;
switch(x)
{
case 0: break;
case 1:
n1=n1+1;
c1=c1+50;
break;
```

case 2:

$n2=n2+1;$

$c2=c2+150;$

break;

case 3:

$n3=n3+1;$

$c3=c3+200;$

break;

case 4:

$n4=n4+1;$

$c4=c4+100;$

break;

case 5:

$n5=n5+1;$

$c5=c5+170;$

break;

case 6:

$n6=n6+1;$


```

c6=c6+70;

break;

case 7:

delete1();

break;

}

}while(x>=1&& x<=7);

getch();

}

void souvenirs::delete1()

{

int y,z;

cout<<"Do you want to delete any item?"<<endl;

cout<<"(Press 0 if no and 1 if yes and 7 after
deleting all items)"<<endl;

cin>>y;

if(y==0)

bill();

```

```
if(y==1)
{
cout<<"Which item do you want to
delete?"<<endl;
do
{
cin>>z;
switch(z)
{
case 0:
break;
case 1:
n1=n1-1;
c1=c1-50;
break;
case 2:
n2=n2-1;
c2=c2-150;
```

break;

case 3:

$n3=n3-1$;

$c3=c3-200$;

break;

case 4:

$n4=n4-1$;

$c4=c4-100$;

break;

case 5:

$n5=n5-1$;

$c5=c5-170$;

break;

case 6:

$n6=n6-1$;

$c6=c6-70$;

break;

case 7:

```

bill();

break;

}

}while(z>=1&& z<=7);

}

getch();

}

void souvenirs::bill()

{

int c;

cout<<"\t"<<"SMVP Park"<<"\t"<<endl;

cout<<"\t"<<"SMVP Shops Bill"<<"\t"<<endl;

cout<<"\t"<<"Item"<<"\t"<<"Quantity"<<"\t"<<"

Cost"<<"\t"<<endl;

if(n1>0)

cout<<"\t"<<"Hat

:"<<"\t"<<n1<<"\t"<<c1<<"\t"<<endl;

if(n2>0)

```

```

cout<<"\t"<<"Mug
:"<<"\t"<<n2<<"\t"<<c2<<"\t"<<endl;

if(n3>0)

cout<<"\t"<<"Large bag
:"<<"\t"<<n3<<"\t"<<c3<<"\t"<<endl;

if(n4>0)

cout<<"\t"<<"Photo
:"<<"\t"<<n4<<"\t"<<c4<<"\t"<<endl;

if(n5>0)

cout<<"\t"<<"Photo Frame
:"<<"\t"<<n5<<"\t"<<c5<<"\t"<<endl;

if(n6>0)

cout<<"\t"<<"Bracelet
:"<<"\t"<<n6<<"\t"<<c6<<"\t"<<endl;

cout<<"Total amount
="<<c1+c2+c3+c4+c5+c6<<endl;

cout<<"Bill amount ="<<c1+c2+c3+c4+c5+c6<<"
+ 18% (GST
TAX)="<<c1+c2+c3+c4+c5+c6+((c1+c2+c3+c4+
c5+c6)*18/100)<<endl;

```

```
total=c1+c2+c3+c4+c5+c6+((c1+c2+c3+c4+c5+c6)*18/100);
```

```
getch();
```

```
return;
```

```
}
```

```
class restaurant
```

```
{
```

```
private:
```

```
int
```

```
n1,n2,n3,v1,v2,v3,cn1,cn2,cn3,cv1,cv2,cv3,total1;
```

```
public:
```

```
int amount2()
```

```
{
```

```
return total1;
```

```
}
```

```
restaurant()
```

```
{
```

```
n1=0;
```

```
n2=0;
n3=0;
v1=0;
v2=0;
v3=0;
cv1=0;
cv2=0;
cv3=0;
cn1=0;
cn2=0;
cn3=0;
}
void getdata4();
void delete4();
void bill4();
}r;
void restaurant::getdata4()
{
```

```
int x;

cout<<"/t"<<"WELCOME TO SMVP
PARKS"<<"/t"<<endl;

cout<<"/t"<<"SMVP
RESTAURANT"<<"/t"<<endl;

cout<<"MENU"<<endl;

cout<<"1.Dosa with Sambar"<<endl;
cout<<"2.Veg biriyani"<<endl;
cout<<"3.Pulao with Channa gravy"<<endl;
cout<<"4.Chapathi with chicken gravy"<<endl;
cout<<"5.Mutton biriyani"<<endl;
cout<<"6.Steamed rice with fish gravy"<<endl;
cout<<"7.Exit"<<endl;

do
{
cout<<"Enter choice"<<endl;

cin>>x;

switch(x)
```



```
{  
case 0:  
break;  
case 1:  
v1++;  
cv1=cv1+70;  
break;  
case 2:  
v2++;  
cv2=cv2+170;  
break;  
case 3:  
v3++;  
cv3=cv3+150;  
break;  
case 4:  
n1++;  
cn1=cn1+160;
```

```
break;

case 5:

n2++;

cn2=cn2+180;

break;

case 6:

n3++;

cn3=cn3+200;

break;

case 7:

delete4();

break;

}

}while(x>=1&& x<=7);

getch();

}

void restaurant::delete4()

{
```

```
int c;

cout<<"Do you want to delete any item?"<<endl;
cout<<"(Enter 1 for yes and 0 for no)"<<endl;
cin>>c;

if(c==0)
bill4();

if(c==1)
{
cout<<"Enter item to delete and then 7 to
exit"<<endl;
do
{
cin>>c;
switch(c)
{
case 0:
break;
case 1:
```

```
v1--;  
cv1=cv1-70;  
break;  
case 2:  
v2--;  
cv2=cv2-170;  
break;  
case 3:  
v3--;  
cv3=cv3-150;  
break;  
case 4:  
n1--;  
cn1=cn1-160;  
break;  
case 5:  
n2--;  
cn2=cn2-180;
```

```

break;

case 6:

n3--;

cn3=cn3-200;

break;

case 7:

bill4();

break;

}

}while(c>=1&& c<=7);

}

getch();

}

void restaurant::bill4()

{

cout<<endl;

cout<<"\t"<<"WELCOME TO SMVP

PARKS"<<"\t"<<endl;

```

```

cout<<"\t"<<"SMVP
RESTAURANT"<<"\t"<<endl;

cout<<"\t"<<"ITEM:"<<"\t"<<"COST:"<<"\t"<<e
ndl;

if(v1>0)

cout<<"\t"<<"Dosa with
sambar"<<"\t"<<cv1<<"\t"<<endl;

if(v2>0)

cout<<"\t"<<"Veg
biriyani"<<"\t"<<cv2<<"\t"<<endl;

if(v3>0)

cout<<"\t"<<"Steamed rice with Channa
curry"<<"\t"<<cv3<<"\t"<<endl;

if(n1>0)

cout<<"\t"<<"Chapathi with chicken
curry"<<"\t"<<cn1<<"\t"<<endl;

if(n2>0)

cout<<"\t"<<"Mutton
biriyani"<<"\t"<<cn2<<"\t"<<endl;

```

```

if(n3>0)

cout<<"\t"<<"Steamed rice with fish
curry"<<"\t"<<cn3<<"\t"<<endl;

cout<<"Amount
total:"<<cv1+cv2+cv3+cn1+cn2+cn3<<endl;

cout<<"With Tax ( GST 18%
):"<<cv1+cv2+cv3+cn1+cn2+cn3+((cv1+cv2+cv3
+cn1+cv2+cv3)*18/100)<<endl;

cout<<"Thank You for visiting us"<<endl;

total1=cv1+cv2+cv3+cn1+cn2+cn3+((cv1+cv2+c
v3+cn1+cn2+cn3)*18/100);

getch();
}

class pticket
{
private:

int h1,h2,m1,m2,h3,m3,total2;

public:

int amount3()

```

```

{
return total2;
}

void getdata1();
void printdata();
}t;

//Where pticket1 represents Parking ticket and
h=hours,m=minutes//

void pticket::getdata1()
{
cout<<"\t"<<"SMVP PARKS"<<"\t"<<endl;
cout<<"\t"<<"Parking Ticket"<<"\t"<<endl;
cout<<"Enter the initial time"<<endl;
cin>>h1>>m1;
cout<<"Enter the final time"<<endl;
cin>>h2>>m2;
h3=h2-h1;
m3=m2-m1;

```



```

if(m3>=60)
h3++;
}
void pticket::printdata()
{
cout<<"\t"<<"WELCOME TO SMVP
PARKS"<<"\t"<<endl;
cout<<"\t"<<"Parking ticket"<<"\t"<<endl;
cout<<"Time of
Entrance:"<<h3<<":"<<m3<<endl;
cout<<"Amount=Rs.50"<<endl;
cout<<"Note : Extra cash (Rs.20 per hour) has to
be paid if parked for time exceeding the time limit
( 6 hours )"<<endl;
if(h3>=6)
total2=50;
else
total2=0;
getch();

```

```
}  
class clothing  
{  
private:  
int n1,n2,n3,n4,n5,n6,c1,c2,c3,c4,c5,c6,total3;  
public:  
int amount4()  
{  
return total3;  
}  
void getdata2();  
void delete2();  
void bill2();  
}c;  
void clothing::getdata2()  
{  
int g;
```

```
cout<<"\t"<<"WELCOME TO SMVP  
PARKS"<<"\t"<<endl;  
cout<<"\t"<<"SMVP Clothing"<<"\t"<<endl;  
cout<<"1.Kids shirt girls"<<endl;  
cout<<"2.Kids shirt boys"<<endl;  
cout<<"3.Kids trousers"<<endl;  
cout<<"4.Kids skirt"<<endl;  
cout<<"5.Women dress"<<endl;  
cout<<"6.Men pants"<<endl;  
cout<<"7.Exit"<<endl;  
do  
{  
cout<<"Enter choice"<<endl;  
cin>>g;  
switch(g)  
{  
case 0:  
break;
```

case 1:

n1++;

c1=c1+190;

break;

case 2:

n2++;

c2=c2+190;

break;

case 3:

n3++;

c3=c3+210;

break;

case 4:

n4++;

c4=c4+210;

break;

case 5:

n5++;

```

c5=c5+350;

break;

case 6:

n6++;

c6=c6+350;

break;

case 7:

delete2();

break;

}

}while(g>=1&&g<=7);

getch();

}

void clothing::delete2()

{

int o;

cout<<"Do you want to delete some item? (Press 1
if no and 0 if yes and then 7 to exit)"<<endl;

```

```
cin>>o;
if(o==1)
bill2();
if(o==0)
{
switch(o)
{
case 0:
break;
case 1:
n1=n1-1;
c1=c1-190;
break;
case 2:
n2=n2-1;
c2=c2-190;
break;
case 3:
```

```
n3=n3-1;  
c3=c3-210;  
break;  
case 4:  
n4=n4-1;  
c4=c4-210;  
break;  
case 5:  
n5=n5-1;  
c5=c5-350;  
break;  
case 6:  
n6=n6-1;  
c6=n6-350;  
break;  
case 7:  
bill2();  
break;
```

```

}

}

getch();

}

void clothing::bill2()

{

cout<<"\t"<<"WELCOME TO SMVP
PARKS"<<"\t"<<endl;

cout<<"\t"<<"SMVP CLOTHING
BILL "<<"\t"<<endl;

cout<<"\t"<<"Item:"<<"\t"<<"Quantity:"<<"\t"<<
"Cost:"<<"\t"<<endl;

if(n1>0)

cout<<"\t"<<"Kids shirt
girls"<<"\t"<<n1<<"\t"<<c1<<"\t"<<endl;

if(n2>0)

cout<<"\t"<<"Kids shirt
boys"<<"\t"<<n2<<"\t"<<c2<<"\t"<<endl;

if(n3>0)

```



```

cout<<"\t"<<"Kids
trousers"<<"\t"<<n3<<"\t"<<c3<<"\t"<<endl;

if(n4>0)

cout<<"\t"<<"Kids
skirt"<<"\t"<<n4<<"\t"<<c4<<"\t"<<endl;

if(n5>0)

cout<<"\t"<<"Women
dress"<<"\t"<<n5<<"\t"<<c5<<"\t"<<endl;

if(n6>0)

cout<<"\t"<<"Men
pants"<<"\t"<<n6<<"\t"<<c6<<"\t"<<endl;

cout<<"Amount total =
"<<c1+c2+c3+c4+c5+c6<<" + GST TAX (18%) =
"<<c1+c2+c3+c4+c5+c6+((c1+c2+c3+c4+c5+c6)
*(18/100))<<endl;

total3=c1+c2+c3+c4+c5+c6+((c1+c2+c3+c4+c5+
c6)*18/100);

getch();

return;

}

```

```
class parks
{
private:
int n,total4;
public:
int amount5()
{
return total4;
}
void input()
{
cout<<"Enter the number of people\n";
cin>>n;
}
void output()
{
cout<<"Welcome to smvp parks/n";
cout<<"Enjoy your ride\n";
```

```

total4=50*n;
}
}p;
void write()
{
ofstream ofile ("parks.dat",ios::app);
i.amount(x1,x2,x3,x4,x5);
ofile.write((char*)&i,sizeof(i));
ofile.close();
}
void main()
{

do{
clrscr();
cout<<"\t"<<"SMVP PARKS"<<"\t"<<endl;
cout<<"1.Parking ticket"<<endl;

```

```
cout<<"2.Souvenirs shop"<<endl;
cout<<"3.Clothing shop"<<endl;
cout<<"4.Restaurant"<<endl;
cout<<"5.Parks"<<endl;
cout<<"6. write in to file\n";
cout<<"7.Display data from file"<<endl;
cout<<"8.Exit"<<endl;
cout<<"Enter choice"<<endl;
cin>>a;
switch(a)
{
case 0:
break;
case 1:
i.getinfo();
t.getdata1();
t.printdata();
x1=t.amount3();
```

```
break;

case 2:

s.getdata();
x2=s.amount1();

break;

case 3:

c.getdata2();
x3=c.amount4();

break;

case 4:

r.getdata4();
r.delete4();
r.bill4();
x4=r.amount2();

break;

case 5:

p.input();
p.output();
```

```
x5=p.amount5();  
break;  
break;  
case 6:  
write();  
break;  
case 7:  
ifstream ifile("parks.dat");  
while(ifile.read((char*)&i,sizeof(i)))  
i.showdetails();  
ifile.close();  
getch();  
break;  
case 8:  
exit(0);  
break;  
}  
}while(a>=1&&a<=8);
```

```
getch();  
}
```


CONCLUSION

I would like to conclude that by introducing such kind of software the manual work has been reduced to a great extent and also there is improvement in programming courses.

While doing this project, I got to know how the present generation is using the technology effectively and efficiently for all programming courses.

The further improvement that I would like to make in this project is to the upcoming entrepreneurs for easy management and handling of the theme park.

REFERENCES AND BIBLIOGRAPHY

1. Class 12 Computer Science Text book by Sumita Arora, Dhanpath Rai Publications New Delhi.
2. Class notes of Computer Science.