

Assignment 1

DRY-RUN EXERCISES

6.1 What will happen when the following program is

```
#include<iostream>
using namespace std;
class construct
```

163

Constructors and Destructors

```
{
    int p,q;
    public:
    construct(int x,int y)
    {
        p=x;
        q=y;
    }
    void Display()
    {
        cout<<p<<"\n"<<q<<"\n";
    }
};

int main()
{
    construct item1(10,20), item2= construct(30,40);
    item1.Display();
    item2.Display();
    return 0;
}
```

6.2 What will happen when the following program is run?

```
#include <iostream>
using namespace std;
class A
{
    int x;
    public:
    A(int d)
    {
        x=d;
    }
    ~A()
    {
        cout<<x;
    }
};

int main()
{
    A a(50), b(60);
    for(int i=1; i<=10; i++)
    {
        A y(i);
    }
    A c(70);
    return 0;
}
```

6.3 What will happen when the following program is run?

```
#include <iostream>
using namespace std;
class pract
{
    int a,b,c;
public:
    pract(int x,int y,int z)
    {
        a=x;
        b=y;
        c=z;
    }
    void show()
    {
        cout<<"\nValues "<<a<<"\t"<<b<<"\t"<<c;
    }
};

int main()
{
    pract Object(10,20,30);
    pract Object2=Object;
    pract Object3=Object2;
    Object3.show();
    return 0;
}
```

6.4 What will happen when the following program is run?

```
#include <iostream>
using namespace std;
class demo
{
    static int cnt;
    int data;
public:
    demo(int d)
    {
        data = d;
        cnt++;
    }
    ~demo()
    {
        cnt--;
    }
    void show()
    {
        cout<<"\nValues "<<data<<"\t"<<cnt;
    }
    static void fun(demo * & ob)
}
```

```
{
    demo temp(ob.data);
    temp.show();
    ob.show();
}

};
int demo::cnt;
int main()
{
    demo ob1(20);
    ob1.show();
    demo::fun(ob1);
    ob1.show();
    demo ob2(30);
    ob1.show();
    ob2.show();
    return 0;
}
```

Assignment 2

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher, and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise "Required copies not in stock" is displayed. Design a system using a class called books with suitable member functions and constructors. Use new operator in constructors to allocate memory space required.

```
#include <iostream>
using namespace std;

class book
{
    char **author;
    char **title;
    float *price;
    char **publisher;
    int *stock_copy;
    int size;
public:
    book();
    //set size to 4 and create 4 entries of books after allocating memory
    //Example : title[0]="object oriented programming with c++";
    void book_detail(int i); //call from within buy if copies are available
    void buy(int i); //check the index search has returned and ask
    //for n            umber of copies required, if those copies are
    //in stock show total price
    int search(); //return the index of the matching book
};

int main()
{
    book b1;
    int result;
    result = b1.search();
    b1.buy(result);
    return 0;
}

/*output smaple

Enter author name : shahani
and title of book in small latter : computer algorithm
*****book detail *****
```

Title Author Name Stock copy
computer algorithm shahani 80
How many copies of this book is required : ? 78
Total price = 14079 TK
*/

20000			
ptr	1000		
10000	ptr[0]	ptr[1]	ptr[2]
2012			
ptr	1000		
10000	ptr[0]	ptr[1]	ptr[2]
2024			
ptr	1000		
10000	ptr[0]	ptr[1]	ptr[2]
2036			
ptr	1000		
10000	ptr[0]	ptr[1]	ptr[2]
2048			
ptr	1000		
10000	ptr[0]	ptr[1]	ptr[2]

ptrTest ptrTest+1

20000 2012