ST. JOSEPH UNIVERSITY IN TANZANIA

St. Joseph College of Engineering and Technology
Department of Computer Science and Engineering
Object Oriented Programming Laboratory 055CS48
By Robert Innocent Karamagi

List of Experiments

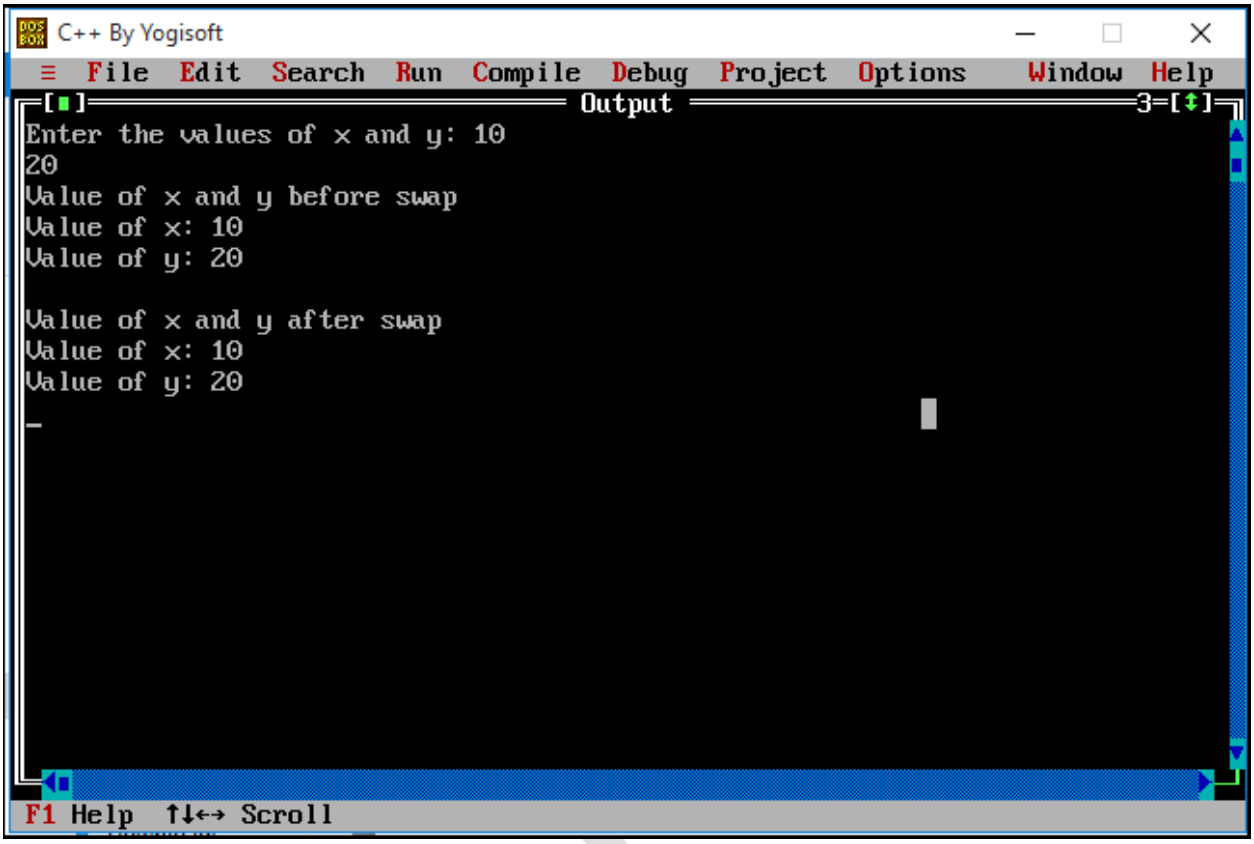
1. (a) Function Call by Value
   (b) Function Call by Reference
   (c) Function Call by Address
2. Function with Default Argument
3. Class with primitive data member
4. Class with array as data member
5. Class with constant data members
6. Class with pointers as data member
7. Static member function
8. Function Overloading
9. Multilevel Inheritance
10. Virtual base class
11. String Manipulation using operator
12. (a) Operator overloading using unary operator
    (b) Operator overloading using binary operator
13. Template function
14. (a) A simple java program with a control structure
    (b) String handling in java
15. Multiple inheritance through interface in java
16. Creating user defined package in java
17. Multithreading in java
18. User defined exception with finally block in java

/*Experiment 1A*/

/*Function call by value*/

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

void swap(int a, int b);
void swap(int a, int b)
{
        int temp;
        temp=a;
        a=b;
        b=temp;
}

void main()
{
        clrscr();
        int x,y;
        cout<<"Enter the values of x and y: ";
        cin>>x>>y;
        cout<<"Value of x and y before swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        swap(x,y);
        cout<<"\nValue of x and y after swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        getch();
}
```

Output



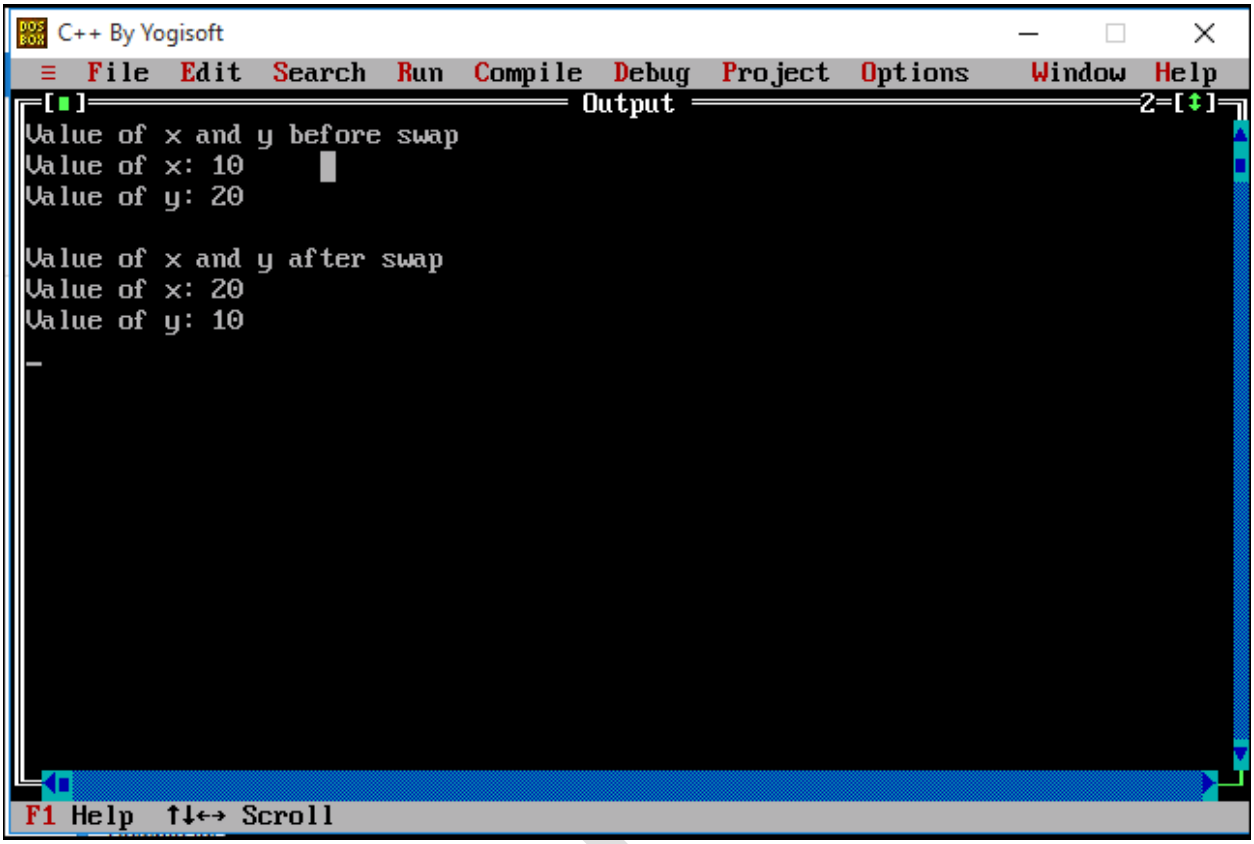
When the above code is put together in a file, compiled and executed, it produces the result which shows that there is no change in the values though they had been changed inside the function. By default, C++ uses call by value to pass the arguments. Thus code within a function cannot alter the arguments used to call the function.

/*Experiment 1B*/

/*Function call by reference*/

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


void swap(int &a, int &b);
void swap(int &a, int &b)
{
        int temp;
        temp=a;
        a=b;
        b=temp;
}

void main()
{
        clrscr();
        int x=10,y=20;
        cout<<"Value of x and y before swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        swap(x,y);
        cout<<"\nValue of x and y after swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        getch();
}
```

Output

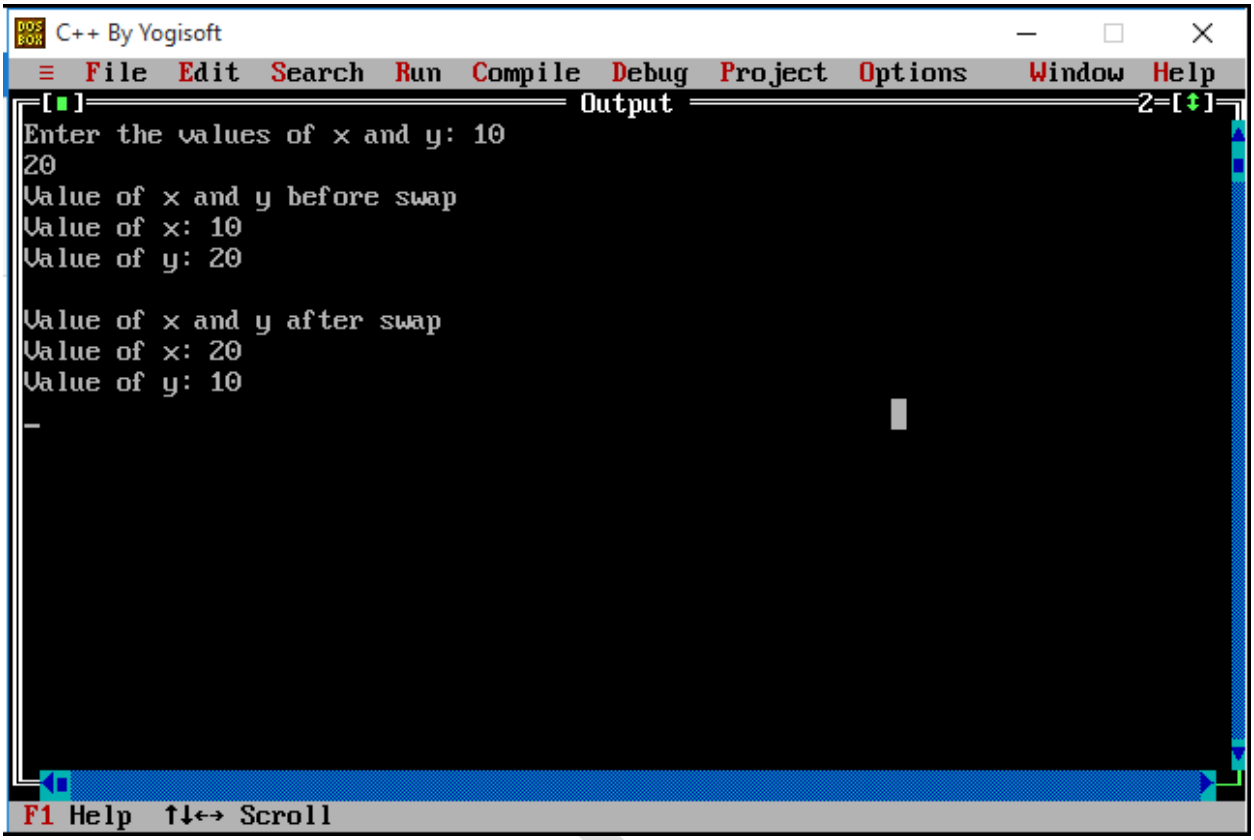


/*Experiment 1C*/

/*Function call by address or pointer*/

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

void swap(int *a, int *b);
void swap(int *a, int *b)
{
        int temp;
        temp=*a;
        *a=*b;
        *b=temp;
}

void main()
{
        clrscr();
        int x,y;
        cout<<"Enter the values of x and y: ";
        cin>>x>>y;
        cout<<"Value of x and y before swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        swap(&x,&y);
        cout<<"\nValue of x and y after swap"<<endl;
        cout<<"Value of x: "<<x<<endl<<"Value of y: "<<y<<endl;
        getch();
}
```

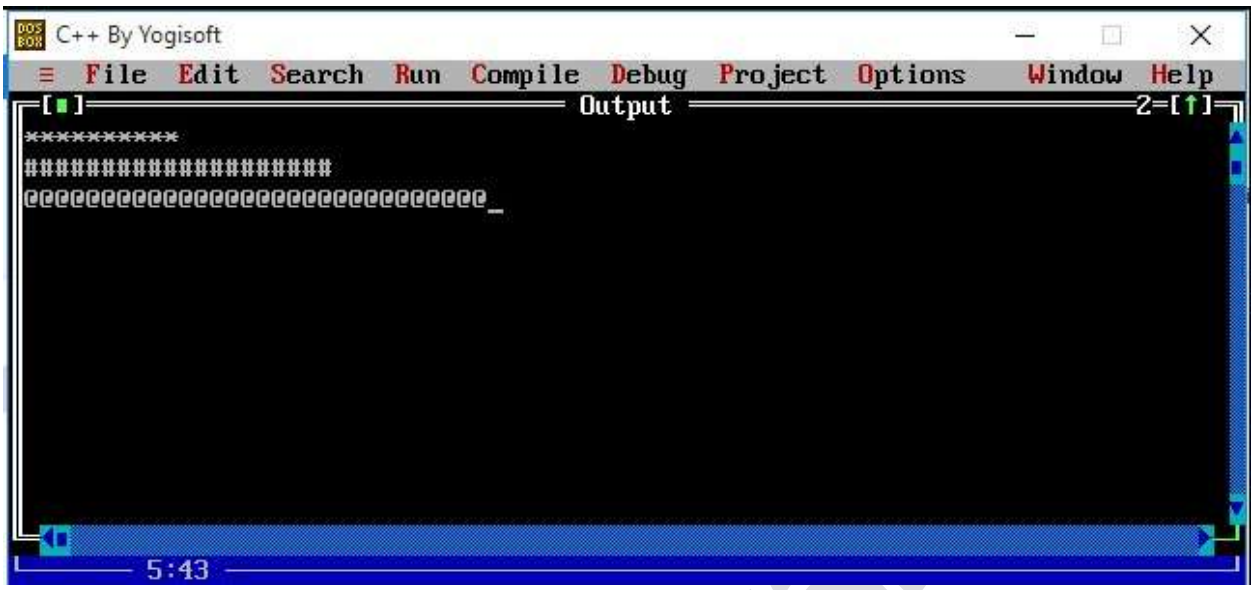
Output



/*Experiment 2/

```
/* Function with default argument */

#include <iostream.h>
#include <conio.h>

void print(char ch= '*', int count= 10);
void print(char ch, int count)
{
        int i;
        for( i = 0; i <count; i++)
        {
                cout<<ch;
        }
}

void main()
{
        clrscr();
        print();
        cout<<"\n";
        print('#',20);
        cout<<"\n";
        print('@',30);
        getch();
}
```

Output

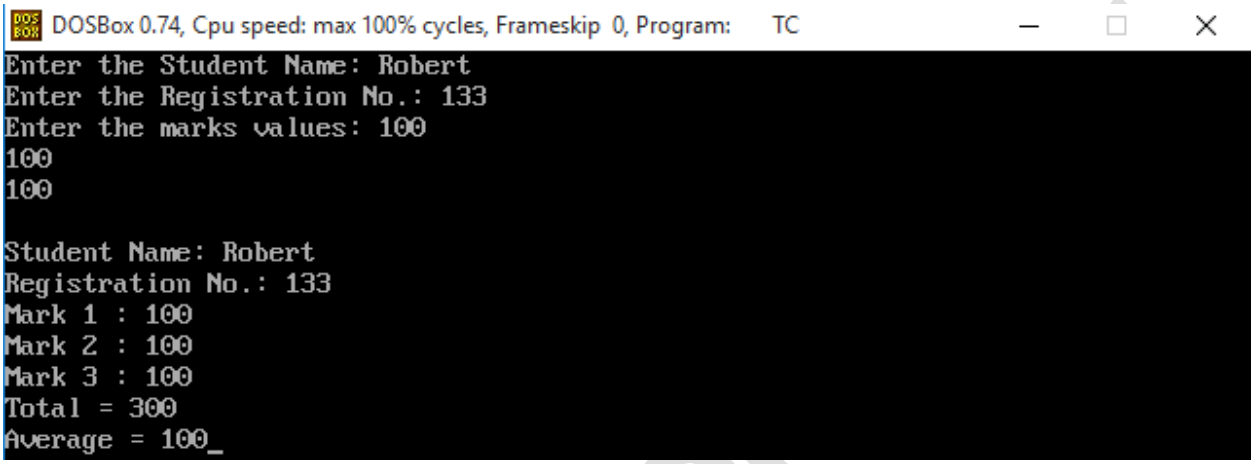


/* Experiment 3*/
/*Class with primitive data member*/

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

class student
{
	public:
		int reg,m1, m2, m3,total;
		char name[20];
		float avg;

		void getdata()
		{
			cout<<"Enter the Student Name: ";
			cin>>name;
			cout<<"Enter the Registration No.: ";
			cin>>reg;
			cout<<"Enter the marks values: ";
			cin>>m1>>m2>>m3;
		}

		void calculation()
		{
			total = m1+m2+m3;
			avg = total/3;
		}

		void display()
		{
			cout<<"\nStudent Name: "<<name;
			cout<<"\nRegistration No.: "<<reg;
			cout<<"\nMark 1 : "<<m1;
			cout<<"\nMark 2 : "<<m2;
			cout<<"\nMark 3 : "<<m3;
			cout <<"\nTotal = "<<total;
			cout <<"\nAverage = "<<avg;
		}
} ;

void main()
{
	clrscr();
	student s;
```

```
        s.getdata();
        s.calculation();
        s.display();
        getch();
}
```

Output



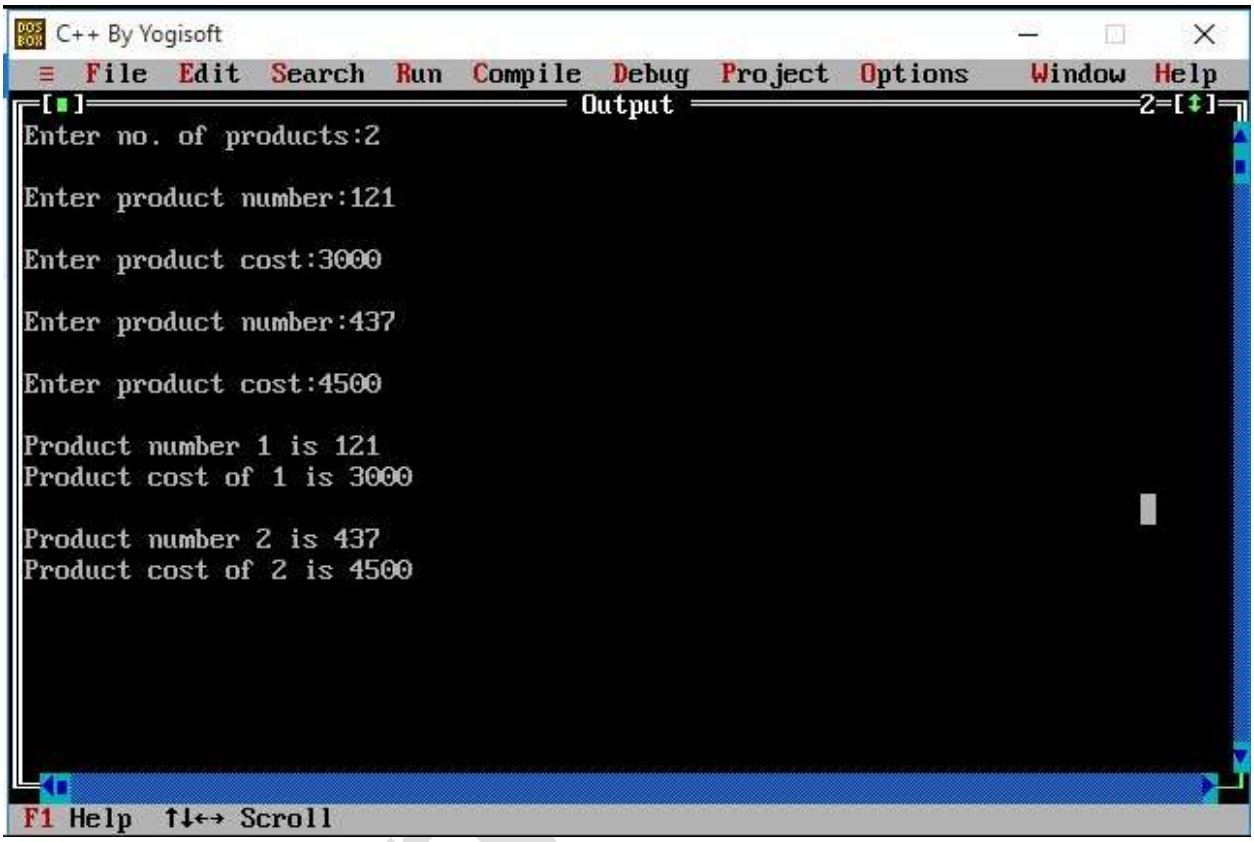
/*Experiment 4*/

/*Classes with array as data member*/

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

class shopping
{
        int pno[10],i,n;
        float pcost[10];

        public:
                void getproduct();
                void displayproduct();
};

void shopping::getproduct()
{
        cout<<"Enter no. of products:";
        cin>>n;
        for (i=1;i<=n;i++)
        {
                cout<<"\nEnter product number:";
                cin >>pno[i];
                cout<<"\nEnter product cost:";
                cin>>pcost[i];
        }
}


void shopping::displayproduct()
{
        for (i=1;i<=n;i++)
        {
                cout<<"\nProduct number "<<i<<" is " <<pno[i];
                cout<<"\nProduct cost of "<<i<<" is "<<pcost[i]<<endl;
        }
}

void main()
{
        clrscr();
        shopping a;
        a.getproduct();
        a.displayproduct();
```

```
    getch();
}
```

Output



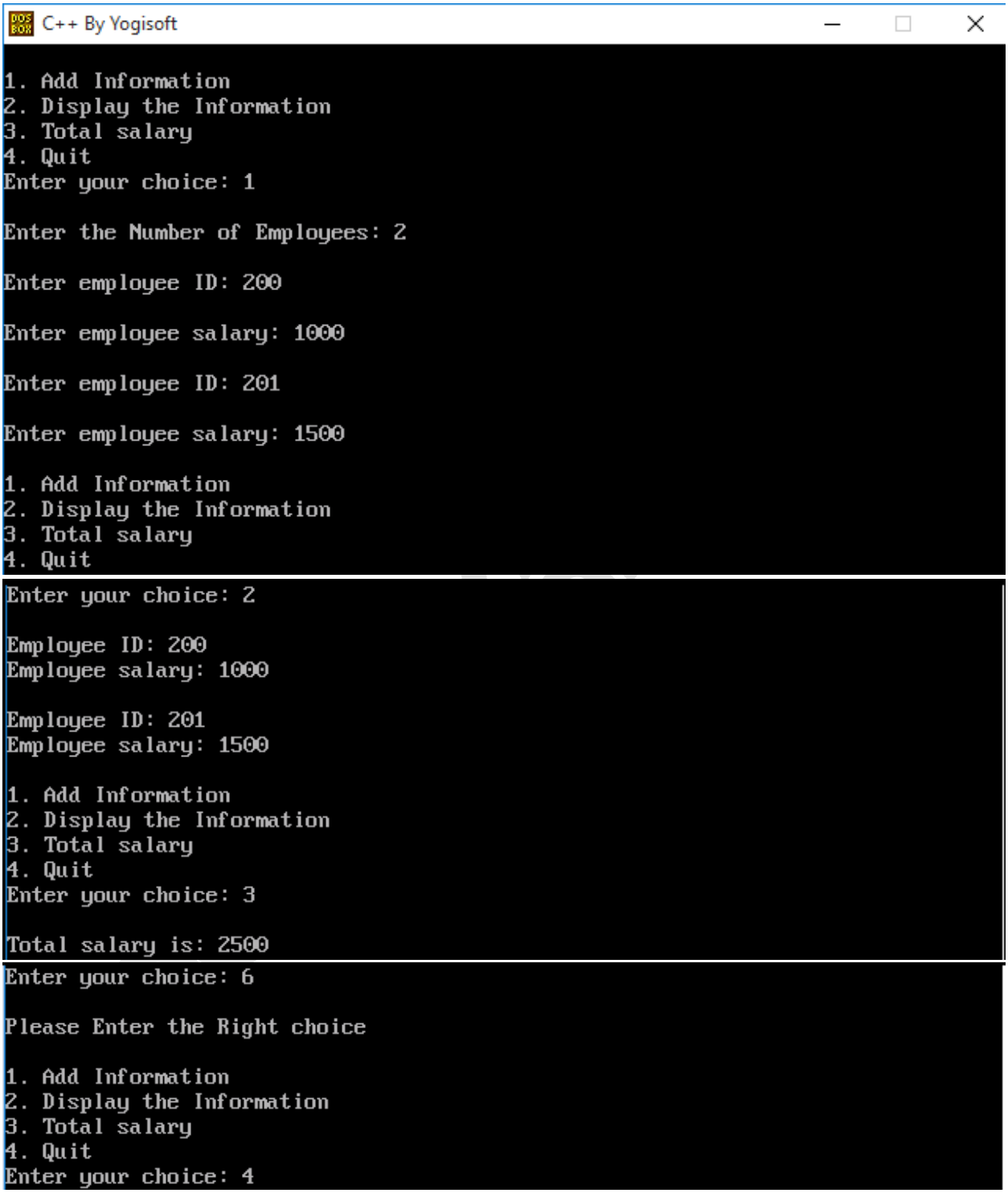
/*Experiment 5*/

/*Classes with constant data member*/

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

const int m=5;
int count = 0;

class employed
{
        int empid[m],n;
        float empsal[m];

        public:
                void input();
                void output();
                void manipulation();
};

void employed::input()
{
        cout<<"\nEnter the Number of Employees: ";
        cin>>n;
        for(count=0;count<n;count++)
        {
                cout<<"\nEnter employee ID: ";
                cin>>empid[count];
                cout<<"\nEnter employee salary: ";
                cin>>empsal[count];
        }
}

void employed::output()
{
        for(count=0;count<n;count++)
        {
                cout<<"\nEmployee ID: "<<empid[count];
                cout<<"\nEmployee salary: "<<empsal[count]<<endl;
        }
}

void employed::manipulation()
{
        int sum=0;
```

```
        for(count=0;count<n;count++)
        {
                sum=sum+empsal[count];
        }
        cout<<"\nTotal salary is: "<<sum<<endl;
 }


int main()
{
        clrscr();
        employed t1;
        int x ;
        do
        {
                cout<<"\n1. Add Information";
                cout<<"\n2. Display the Information";
                cout<<"\n3. Total salary";
                cout<<"\n4. Quit";
                cout<<"\nEnter your choice: ";
                cin>> x;
                switch(x)
                {
                        case 1:
                                t1.input();
                                break;
                        case 2:
                                t1.output();
                                break;
                        case 3:
                                t1.manipulation();
                                break;
                        case 4:
                                return 0;
                        default:
                                cout<<"\nPlease Enter the Right choice "<<endl;
                                break;
                }
        }while(x!=4);
        return 0;

}
```

Output


C++ By Yogisoft
1. Add Information
2. Display the Information
3. Total salary
4. Quit
Enter your choice: 1
Enter the Number of Employees: 2
Enter employee ID: 200
Enter employee salary: 1000
Enter employee ID: 201
Enter employee salary: 1500
1. Add Information
2. Display the Information
3. Total salary
4. Quit
Enter your choice: 2
Employee ID: 200
Employee salary: 1000
Employee ID: 201
Employee salary: 1500
1. Add Information
2. Display the Information
3. Total salary
4. Quit
Enter your choice: 3
Total salary is: 2500
Enter your choice: 6
Please Enter the Right choice
1. Add Information
2. Display the Information
3. Total salary
4. Quit
Enter your choice: 4

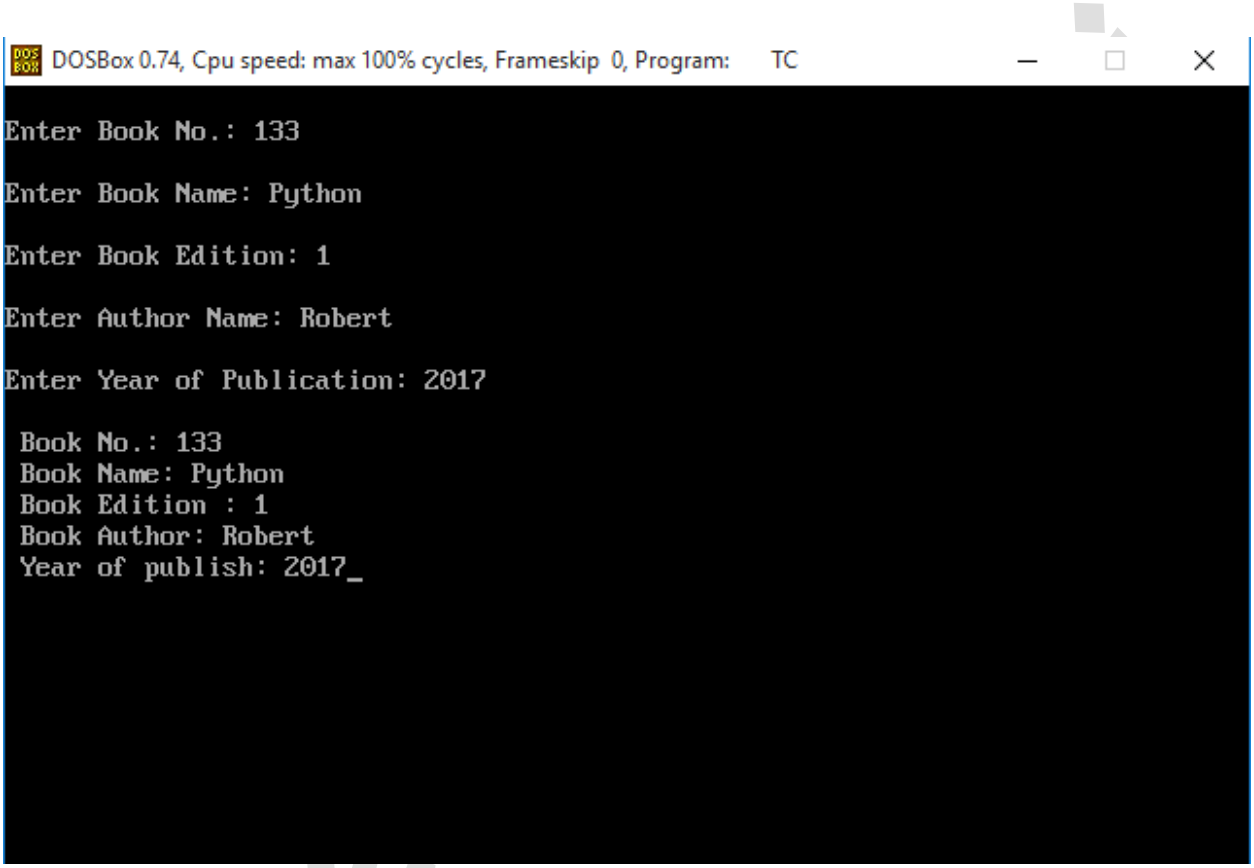
/*Experiment 6*/

/*Classes with pointers as data members*/

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

class library
{
        char *bookauthor, *bookname, *edition;
        int bookid, yrpublish;

        public:
                void information();
                void delivery();
};

void library::information()
{
        bookauthor = new char[10];
        bookname = new char[10];
        edition = new char [10];
        cout<<"\nEnter Book No.: ";
        cin>>bookid;
        cout<<"\nEnter Book Name: ";
        cin>>bookname;
        cout<<"\nEnter Book Edition: ";
        cin>>edition;
        cout<<"\nEnter Author Name: ";
        cin>>bookauthor;
        cout<<"\nEnter Year of Publication: ";
        cin>>yrpublish;
}

void library::delivery()
{
        cout<<"\n Book No.: "<<bookid;
        cout<<"\n Book Name: "<<bookname;
        cout<<"\n Book Edition : "<<edition;
        cout<<"\n Book Author: "<<bookauthor;
        cout<<"\n Year of publish: "<<yrpublish;
}

void main()
{
        clrscr();
```

```
        library l;
        l.information();
        l.delivery();
        getch();
}
```

Output

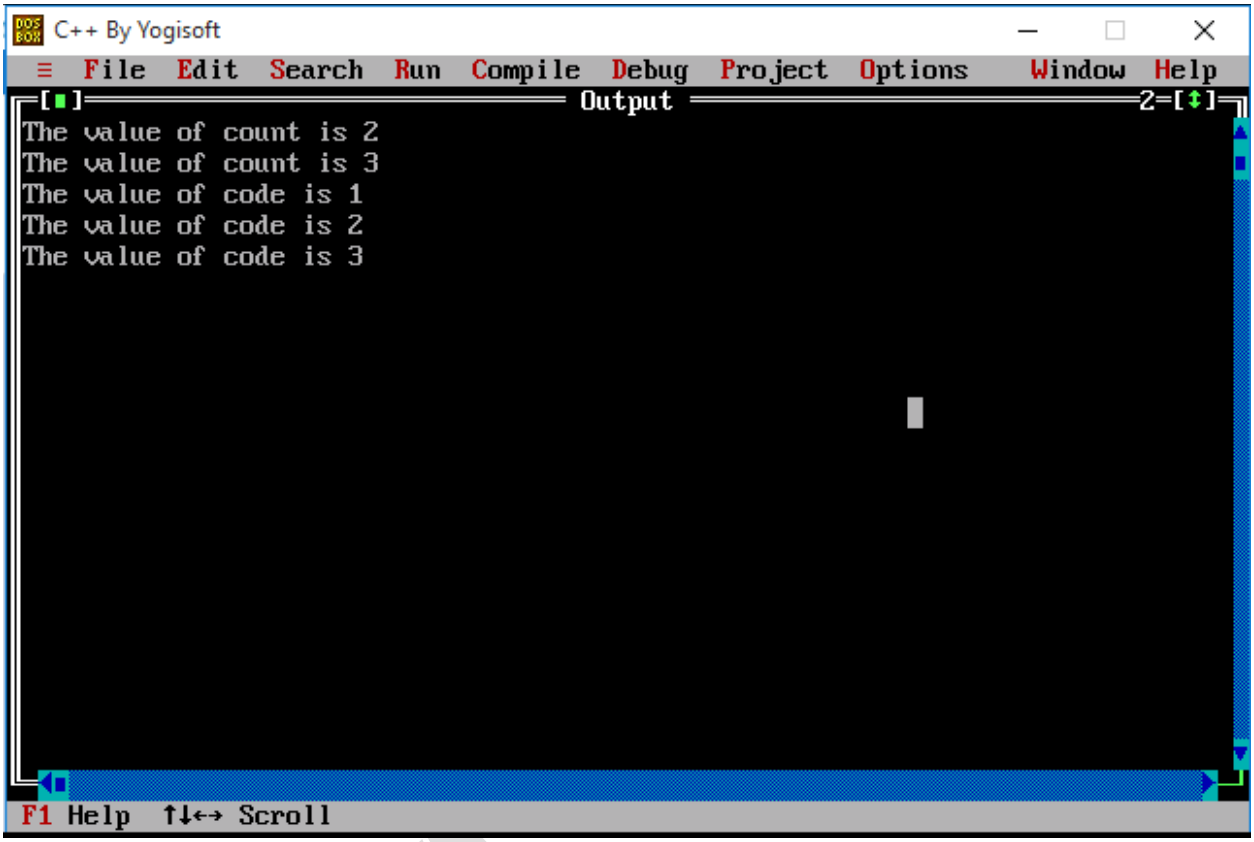


/*Experiment 7*/

```
/*Static Member Function*/

#include<iostream.h>
#include<conio.h>

class test
{
        int code;
        static int count;

        public:
                void sample(void)
                {
                        code=++count;
                }

                void request(void)
                {
                        cout<<"The value of code is "<<code<<endl;
                }

                static void respond(void)
                {
                        cout<<"The value of count is "<<count<<endl;
                }
};

int test::count;

void main()
{
        clrscr();
        test t1,t2;
        t1.sample();
        t2.sample();
        test::respond();   //accessing static function
        test t3;
        t3.sample();
        test::respond();
        t1.request();
        t2.request();
        t3.request();
        getch();
```

}

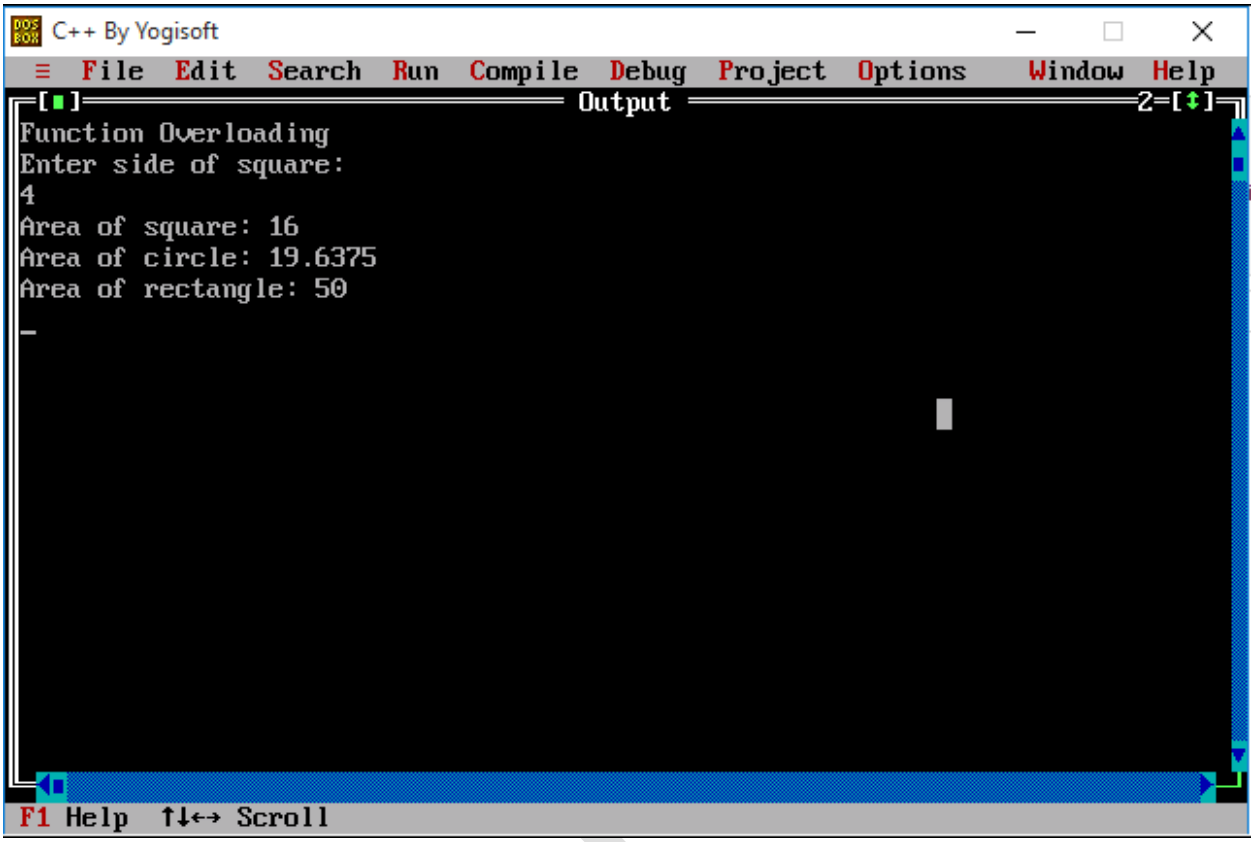
Output



/*Experiment 8 */

```
/*Function overloading*/

#include<iostream.h>
#include<conio.h>

class over
{
	public:
		void area(double r);
		void area(int l, int b);
		void area();
};

void over::area()
{
	int a;
	cout<<"Enter side of square:"<<endl;
	cin>>a;
	cout<<"Area of square: "<<(a*a)<<endl;
}

void over::area(double r)
{
		cout<<"Area of circle: "<<(3.142*r*r)<<endl;
}

void over::area(int l, int b)
{
	cout<<"Area of rectangle: "<<(l*b)<<endl;
}

void main()
{
	clrscr();
	cout<<"Function Overloading"<<endl;
	over o;
	o.area();
	o.area(2.5);
	o.area(5,10);
	getch();
}
```

Output



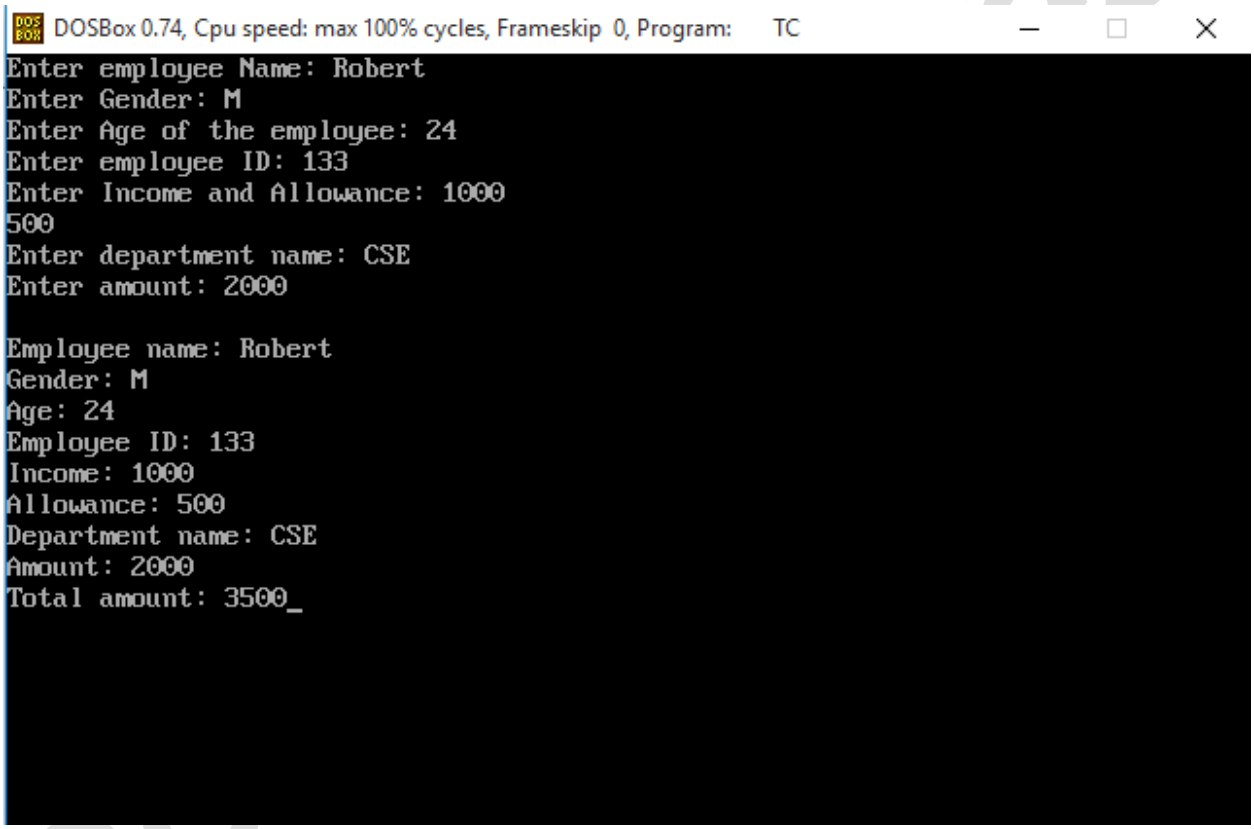
/*Experiment 9*/
/*Multilevel Inheritance*/

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


class company
{
	protected:
		char ename [30],dname [20];
		char gender;
		int age, id;
		float amount,income,allowance,total;

	public:
		void  readdata()
		{
			cout<<"Enter employee Name: ";
			cin>>ename;
			cout<<"Enter Gender: ";
			cin>>gender;
			cout<<"Enter Age of the employee: ";
			cin>>age;
		}

		void writedata()
		{
			cout<<"\n";
			cout<<"Employee name: "<<ename<<endl;
			cout<<"Gender: "<<gender<<endl;
			cout<<"Age: "<<age<<endl;
		}
};

class employee:public company
{

	public:
		void obtain();
		void delivery();
};

void employee::obtain()
{
```

```
	readdata();
	cout<<"Enter employee ID: ";
	cin>>id;
	cout<<"Enter Income and Allowance: ";
	cin>>income>>allowance;
}

void employee::delivery()
{
	writedata();
	cout<<"Employee ID: "<<id<<endl;
	cout<<"Income: "<<income<<endl;
	cout<<"Allowance: "<<allowance<<endl;
}

class dept:public employee
{
	public:

		void receive()
		{
			obtain();
			cout<<"Enter department name: ";
			cin>>dname;
			cout<<"Enter amount: ";
			cin>>amount;
		}

		void display()
		{
			delivery();
			cout<<"Department name: "<<dname<<endl;
			cout<<"Amount: "<<amount<<endl;
		}

		void salary();
};

void dept::salary()
{
	total=amount+income+allowance;
	cout<<"Total amount: "<<total;
}

void main()
```

```
{
        clrscr();
        dept D;
        D.receive();
        D.display();
        D.salary();
        getch();

}
```

Output



/*Experiment 10*/

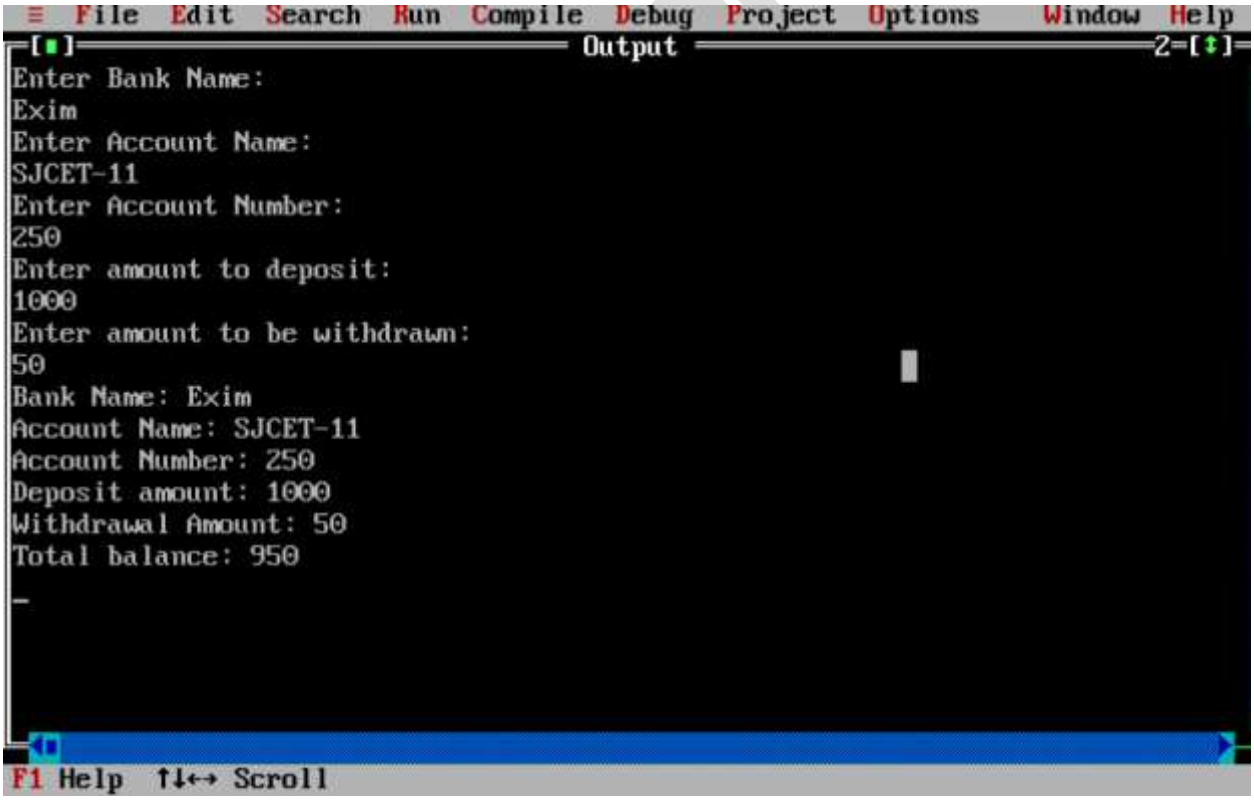
/*Virtual Base Class*/

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

class bank
{
        protected:
        char bname[20], acctname[10];
        int acctno;

        public:
        void take();
        void give();
};

void bank::take()
{
        cout<<"Enter Bank Name: \n";
        cin>>bname;
        cout<<"Enter Account Name: \n";
        cin>>acctname;
        cout<<"Enter Account Number: \n";
        cin>>acctno;
}

void bank::give()
{
        cout<<"Bank Name: "<<bname<<endl;
        cout<<"Account Name: "<<acctname<<endl;
        cout<<"Account Number: "<<acctno<<endl;
}

class deposit:virtual public bank
{
        protected:
                float depamount;

        public:
                void insert();
                void show();
};

void deposit::insert()
```

```
{
	cout<<"Enter amount to deposit: \n";
	cin>>depamount;
}

void deposit::show()
{
	cout<<"Deposit amount: "<<depamount<<endl;
}

class withdraw:virtual public bank
{
	protected:
		float withamount;

	public:
		void collect()
		{
			cout<<"Enter amount to be withdrawn: \n";
			cin>>withamount;
		}

		void distribute()
		{
			cout<<"Withdrawal Amount: "<<withamount<<endl;
		}
};

class balance:public deposit,public withdraw
{
		float totalbal;
		public:
			void calculate();
			void result();
};

void balance::calculate()
{
	totalbal=depamount-withamount;
}

void balance::result()
{
	give();
	show();
```

```
        distribute();
        cout<<"Total balance: "<<totalbal<<endl;

}

void main()
{
        clrscr();
        balance b;
        b.take();
        b.insert();
        b.collect();
        b.calculate();
        b.result();
        getch();

}
```

Output



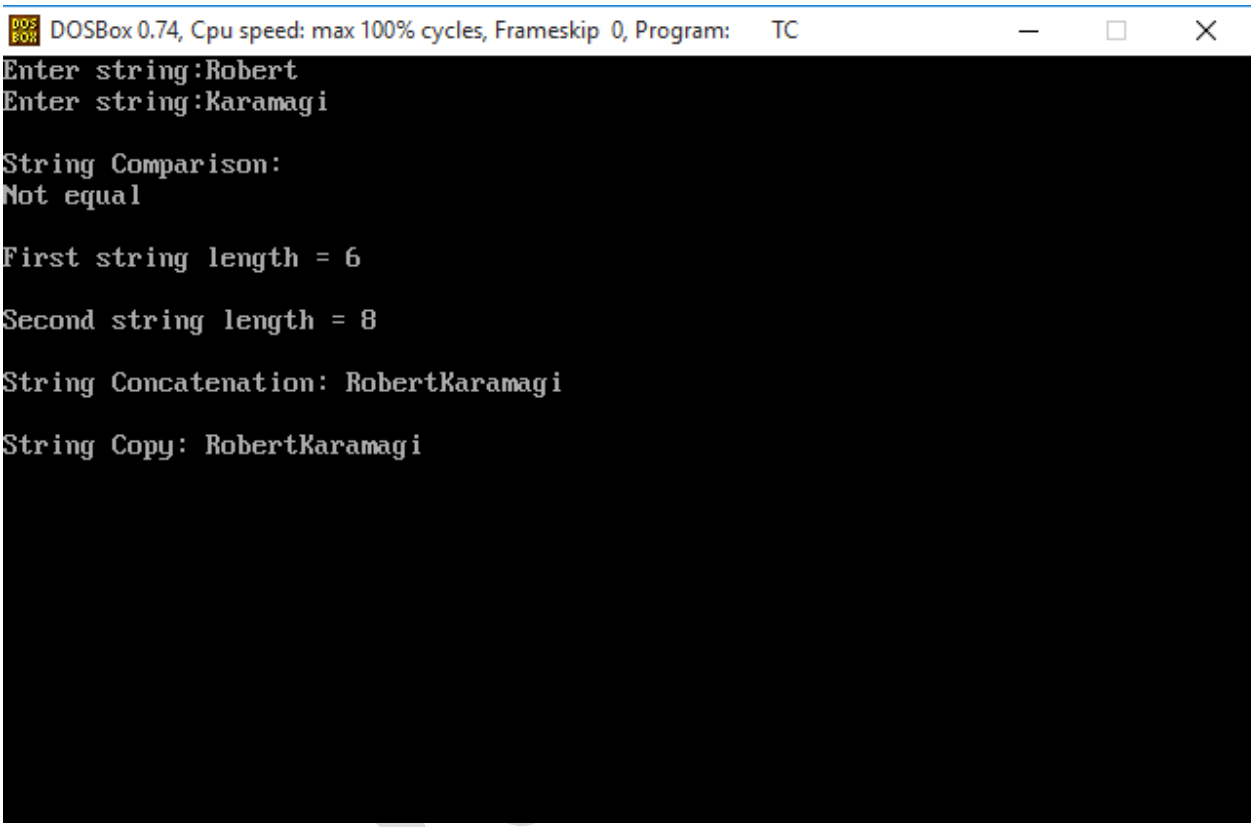
/*Experiment 11*/
/*String Manipulations*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>

class string
{
	protected:
		char s1[25];

	public:
		void getstring();
		void putstring(string&);
		void display();
		string operator+(string&);
		int operator==(string&);
		void operator=(string&);
		void operator+=(string&);
};

void string::getstring()
{
	cout<<"Enter string:";
	gets(s1);
}

void string::putstring(string &s2)
{
	int len=strlen(s2.s1);
	cout<<len;
}

string string::operator+(string &s2)
{
	string temp;
	if( (strlen(s1) + strlen(s2.s1) ) <25 )
	{
		strcpy(temp.s1, s2.s1);
		strcat(temp.s1, s2.s1);
	}
	else
	{
```

```
                cout<<endl<<"Concatenation not possible";
        }
        return temp;
}

int string::operator==(string &s2)
{
        return(strcmp(s1,s2.s1)==0?1:0);
}

void string::operator=(string &s2)
{
        strcpy(s1,s2.s1);
}

void string::operator+=(string &s2)
{
        strcat(s1,s2.s1);
}

void string::display()
{
        cout<<s1;
}

void main()
{
        string b1,b2,b3,b4;
        clrscr();
        b1.getstring();
        b2.getstring();
        cout<<"\nString Comparison:";
        if(b1==b2)
        cout<<endl<<"They are equal";
        else
        cout<<"\nNot equal";
        cout<<endl<<"\nFirst string length = ";
        b1.putstring(b1);
        cout<<endl<<"\nSecond string length = ";
        b2.putstring(b2);
        cout<<endl<<"\nString Concatenation: ";
        b1+=b2;
        b1.display();
        b4=b1;
        cout<<endl<<"\nString Copy: ";
```

```
        b4.display();
        getch();
}
```

Output

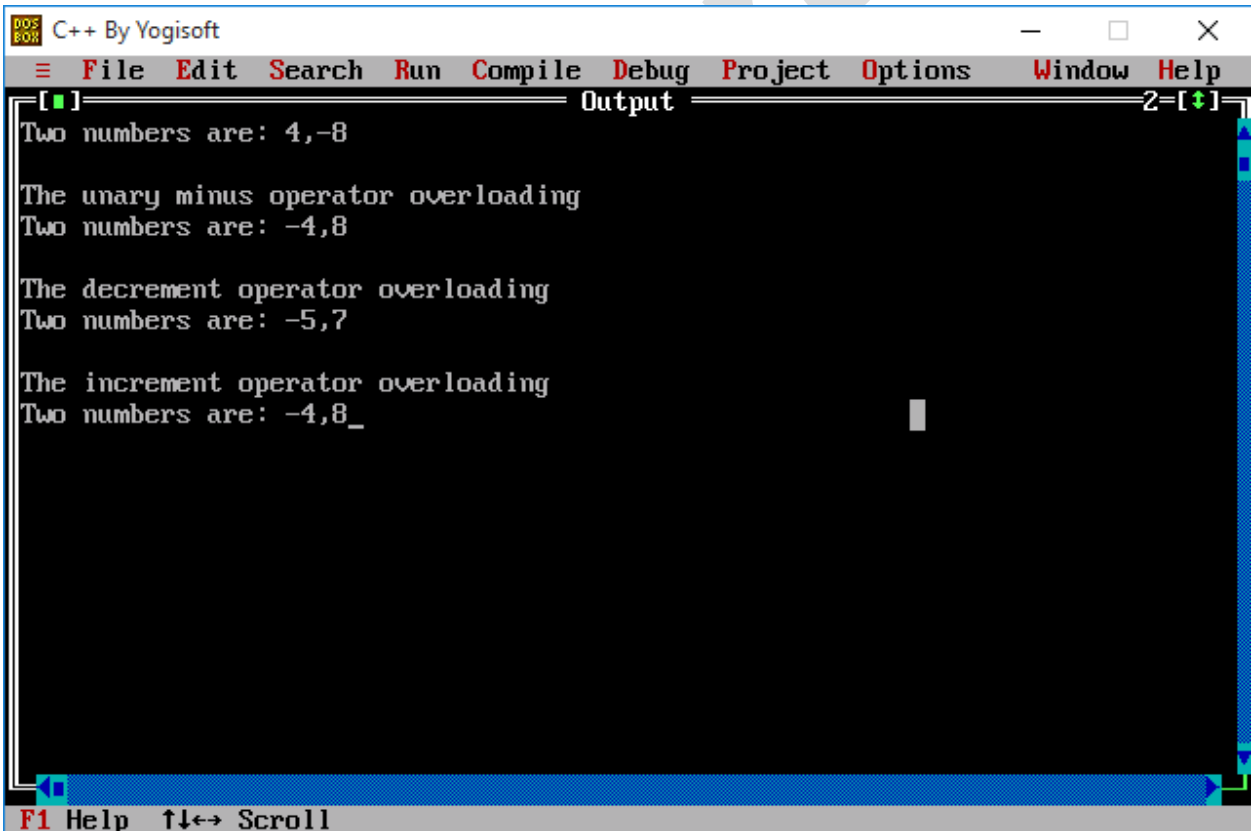


/*Experiment 12A*/

```
/*Operator overloading using unary operator*/

#include <iostream.h>
#include <conio.h>

class unary
{
	private:
		int x,y;

	public:
		unary(int a,int b)
		{
			x=a;
			y=b;
		}

		void display()
		{
			cout<<"Two numbers are: "<<x<<","<<y;
		}

		unary operator-()
		{
			x=-x;
			y=-y;
			return unary(x,y);
		}

		unary operator--()
		{
			x=--x;
			y=--y;
			return unary(x,y);
		}

		unary operator++()
		{
			x=++x;
			y=++y;
			return unary(x,y);
		}
};
```

```
void main()
{
	clrscr();
	unary s(4,-8);
	s.display();
	-s;
	cout<<endl<<endl<<"The unary minus operator overloading"<<endl;
	s.display();
	--s;
	cout<<endl<<endl<<"The decrement operator overloading"<<endl;
	s.display();
	++s;
	cout<<endl<<endl<<"The increment operator overloading"<<endl;
	s.display();
	getch();
}
```

Output



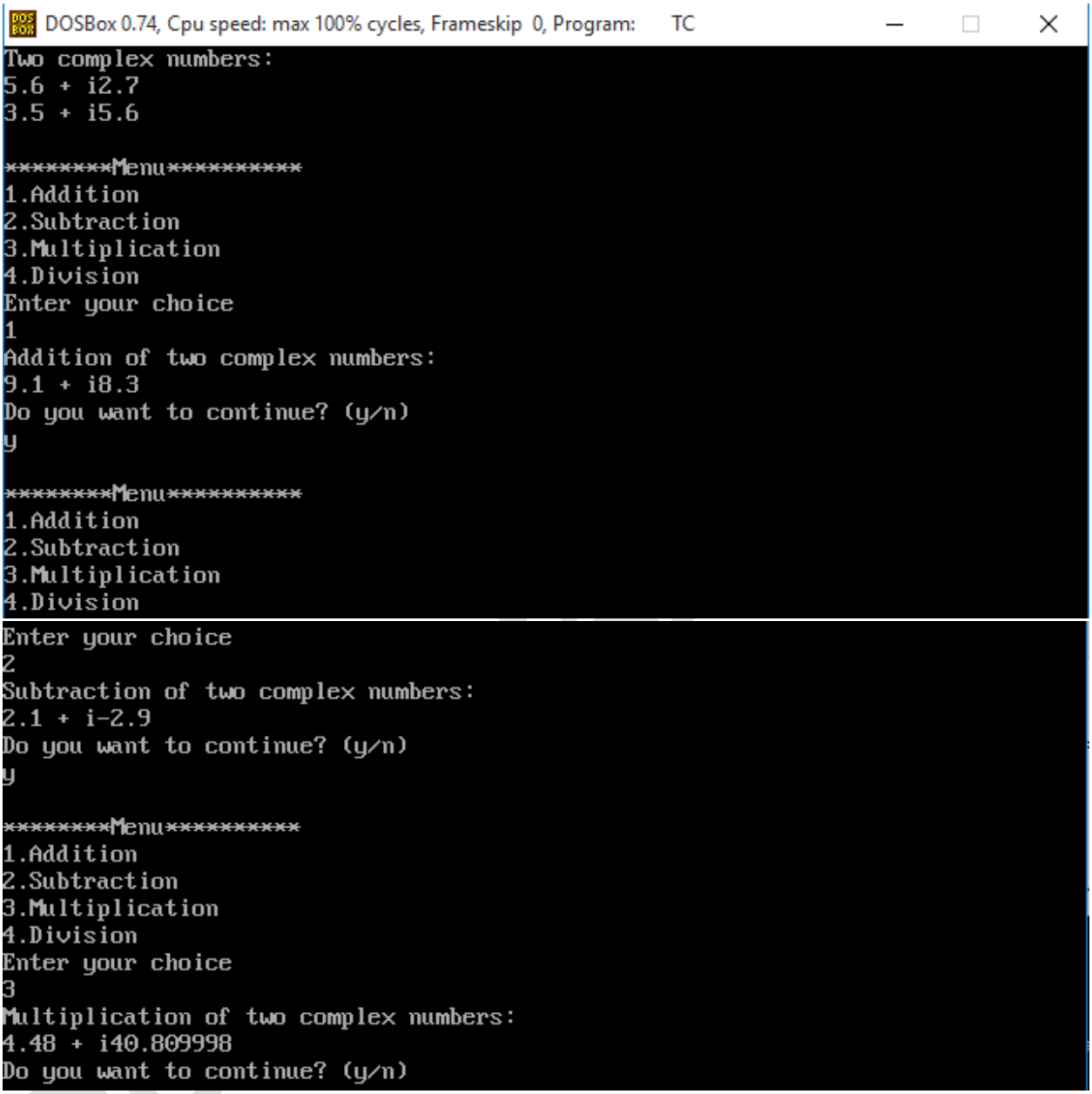
/*Experiment 12B*/

```
/*Operator overloading*/

#include<iostream.h>
#include<conio.h>
#include<math.h>

class complex
{
        float real;
        float imag;

        public:
                complex()
                {
                }

                complex(float x, float y)
                {
                        real=x;
                        imag=y;
                }

                complex operator+ (complex);
                complex operator- (complex);
                complex operator* (complex);
                complex operator/ (complex);

                void display()
                {
                        cout<<real<<" + i"<<imag<<endl;
                }
};

complex complex::operator+ (complex c)
{
        complex c2;
        c2.real=real+c.real;
        c2.imag=imag+c.imag;
        return(c2);
}

complex complex::operator- (complex c)
{
        complex c2;
```

```
	c2.real=real-c.real;
	c2.imag=imag-c.imag;
	return(c2);
}

complex complex::operator* (complex c)
{
	complex c2;
	c2.real=((real*c.real) - (imag*c.imag));
	c2.imag=((real*c.imag) + (imag*c.real));
	return(c2);
}

complex complex::operator/ (complex c)
{
	complex c2;
	c2.real=((real*c.real)+(imag*c.imag)) / (pow(c.real,2)+pow(c.imag,2));
	c2.imag=((imag*c.real)-(real*c.imag)) / (pow(c.real,2)+pow(c.imag,2));
	return(c2);
}
void main()
{
	clrscr();
	complex c1(5.6,2.7), c2(3.5,5.6), c3;
	int op;
	char ch,y,Y;
	cout<<"Two complex numbers: "<<endl;
	c1.display();
	c2.display();
	do
	{
		cout<<endl<<"********Menu**********"<<endl;
		cout<<"1.Addition\n2.Subtraction\n3.Multiplication\n4.Division"<<endl;
		cout<<"Enter your choice"<<endl;
		cin>>op;
		switch(op)
		{
			case 1:
				c3=c1+c2;
				cout<<"Addition of two complex numbers:"<<endl;
				c3.display();
				break;

			case 2:
				c3=c1-c2;
```

```
                cout<<"Subtraction of two complex numbers:"<<endl;
                c3.display();
                break;

            case 3:
                c3=c1*c2;
                cout<<"Multiplication of two complex numbers:"<<endl;
                c3.display();
                break;

            case 4:
                c3=c1/c2;
                cout<<"Division of two complex numbers:"<<endl;
                c3.display();
                break;

            default:
                cout<<endl<<"Aborting!!! Invalid Choice."<<endl;
                break;
        }

        cout<<"Do you want to continue? (y/n)"<<endl;
        cin>>ch;
    }while(ch=='y'||ch=='Y');
    getch();
}
```

Output


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Two complex numbers:
5.6 + i2.7
3.5 + i5.6
********Menu**********
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter your choice
1
Addition of two complex numbers:
9.1 + i8.3
Do you want to continue? (y/n)
y
********Menu**********
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter your choice
2
Subtraction of two complex numbers:
2.1 + i-2.9
Do you want to continue? (y/n)
y
********Menu**********
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter your choice
3
Multiplication of two complex numbers:
4.48 + i40.809998
Do you want to continue? (y/n)

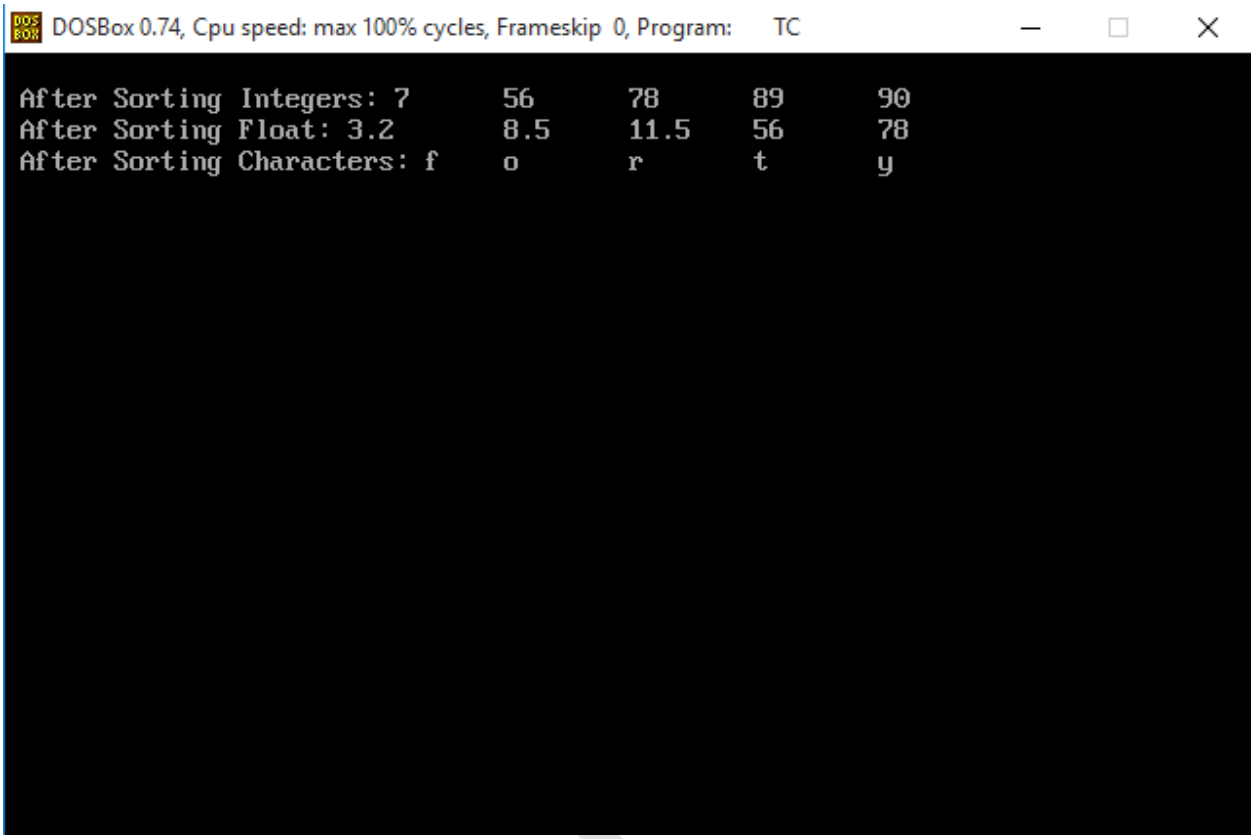
```
y

*********Menu**********
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter your choice
4
Division of two complex numbers:
0.796148 + i-0.502408
Do you want to continue? (y/n)
y

*********Menu**********
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter your choice
7

Aborting!!! Invalid Choice.
Do you want to continue? (y/n)
n
```

/*Experiment 13*/

/*Template Function*/

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

template<class T>
void sort(T a[], int n)
{
	int i, j;
	for(i=0;i<n-1;i++)
	{
		for(j=i+1;j<n;j++)
		{
			if(a[i]>a[j])
			{
				T element;
				element = a[i];
				a[i] = a[j];
				a[j] = element;
			}
		}
	}
}

void main()
{
	clrscr();
	int x[5]={90,7,89,78,56};
	float y[5]={56,78,8.5,11.5,3.2};
	char z[5]={'y','f','t','r','o'};
	sort(x,5);
	cout<<"\n After Sorting Integers: ";
	for(int i=0;i<5;i++)
	cout<<x[i]<<"\t";
	sort(y,5);
	cout<<"\n After Sorting Float: ";
	for(int j=0;j<5;j++)
	cout<<y[j]<<"\t";
	sort(z,5);
	cout<<"\n After Sorting Characters: ";
	for(int k=0;k<5;k++)
	cout<<z[k]<<"\t";
	getch();
}
```

Output




Java
ORACLE

//Experiment 14A
//A simple Java program with control structure

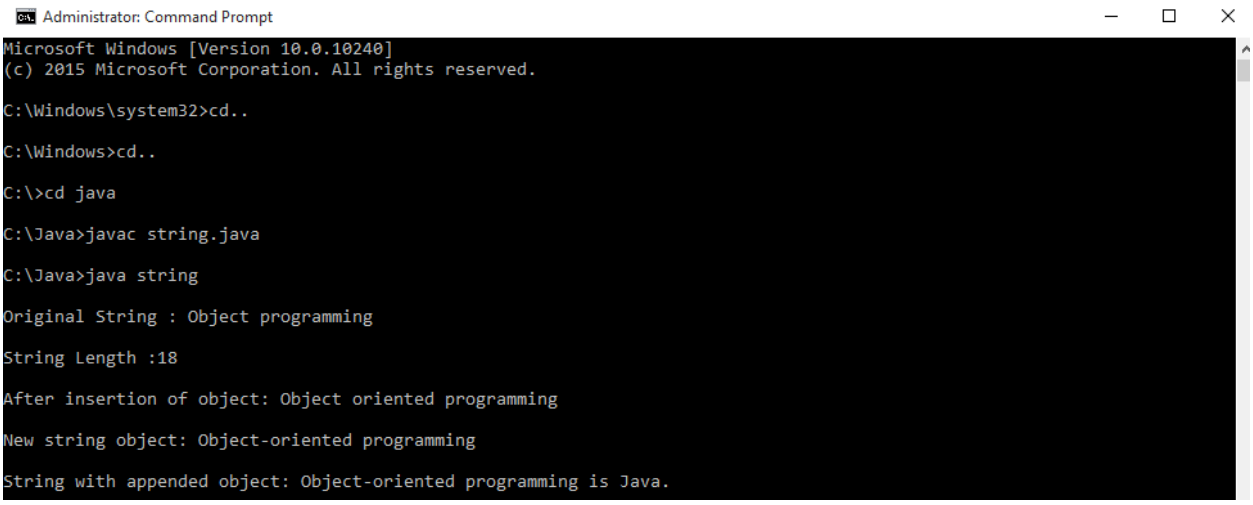
```
import java.io.*;
import java.util.*;

class pyramid
{
	public static void main(String[] args)
	{
		int  i,j;
		for(i=1;i<=5;i++)
		{
			for(j=1;j<=i;j++)
			{
				System.out.print(i);
			}
			System.out.print("\n");
		}
	}
}
```

Output

//Experiment 14B
//String handling in Java

```
import java.io.*;
import java.util.*;
class string
{
	public static void main(String args[])
	{
		StringBuffer str = new StringBuffer("Object programming");
		System.out.println("\nOriginal String : " + str);

		//obtaining string length
		System.out.println("\nString Length :" + str.length());

		//Insertion of string into middle
		str.insert (7, "oriented ");
		System.out.println ("\nAfter insertion of object: " +str);

		//Modifying Characters
		str.setCharAt(6, '-');
		System.out.println("\nNew string object: " +str);

		//Appending a string at the end
		str.append(" is Java.");
		System.out.println("\nString with appended object: " +str);


	}
}
```

Output



//Experiment 15
//Multiple Inheritance through interface in Java

```
import java.io.*;
import java.lang.*;
import java.util.*;

interface Military
{
	int age;
	public void getage(int y)
	{
		age = y;
	}
	public void inspect()
	{
		if( (age>=18) && (age<=50) )
			System.out.println("\nJoin the academy");
		else
			System.out.println("\nYour rejected");
	}
}

interface Hospital
{

	int no_persons;
	public void getperson(int z)
	{
		no_persons = z;
	}
	public void show()
	{
		System.out.println("\nNumber of x-service men in this hospital: " +no_persons);
	}
}

interface Name extends Military, Hospital
{
	public void HM()
	{
		System.out.println("\nCountry information");
		String Hname = "Maweni";
		String MIname = "Burombola JKT";
		System.out.println("\nHospital name: " +Hname);
```

```
            System.out.println("\nMilitary name: " +MIname);
        }
        getage(int y);
        inspect();
        getperson(int z);
        show();

}

class  country implements Name
{
        public static void main(String args[])
        {
                country c = new country();
                c.HM();
                c.getage(50);
                c.getperson(80);
                c.inspect();
                c.show();
        }
}
```

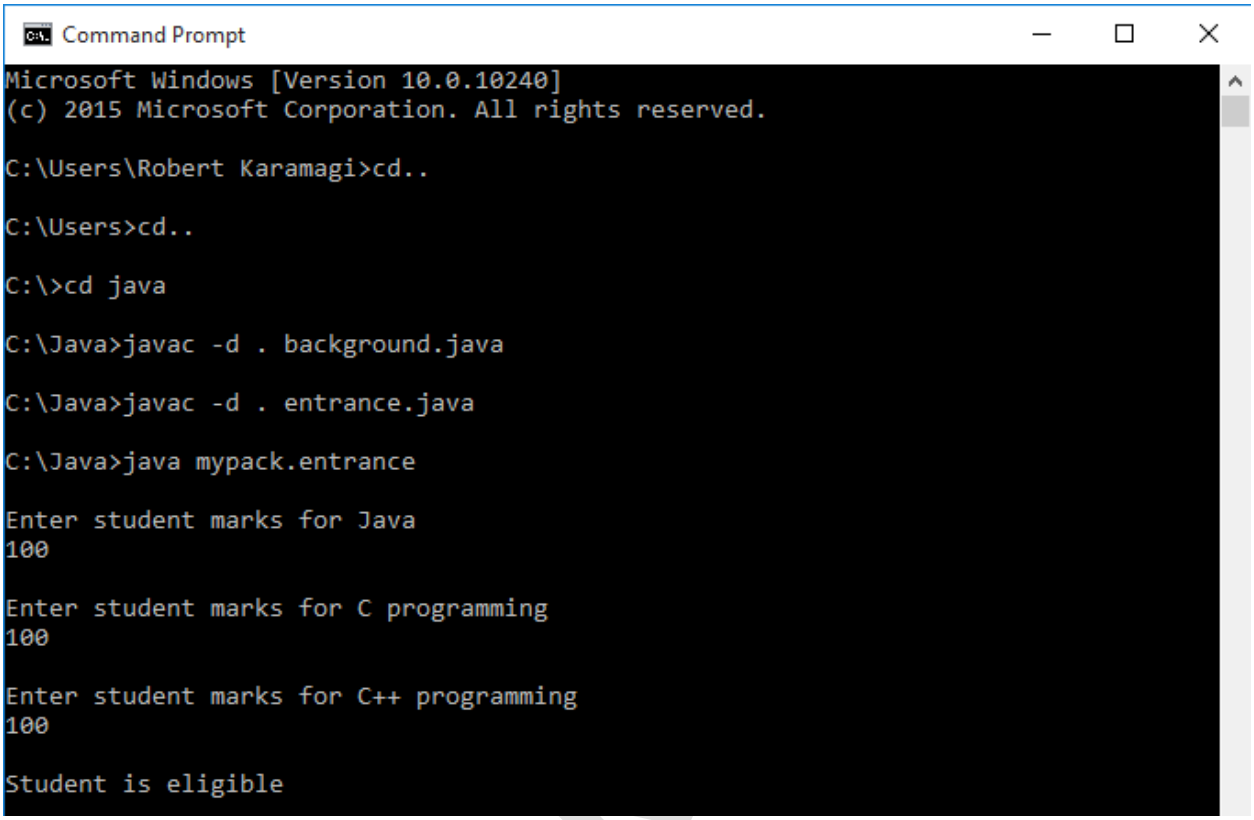
Output

//Experiment 16

//Creating user defined package in Java

```
package quality;
import java.io.*;

public class background
{
	int M1,M2,M3;
	public void testing()throws IOException
	{

		BufferedReader n = new BufferedReader(new InputStreamReader(System.in));
		System.out.println("\nEnter student marks for Java");
		String s1 = n.readLine();
		M1=Integer.parseInt(s1);
		System.out.println("\nEnter student marks for C programming");
		String s2 = n.readLine();
		M2=Integer.parseInt(s2);
		System.out.println("\nEnter student marks for C++ programming");
		String s3 = n.readLine();
		M3=Integer.parseInt(s3);
		if( (M1>=60) && (M2>=60) && (M3>=60) )
			System.out.println("\nStudent is eligible");
		else
			System.out.println("\nNot eligible");
	}
}

package mypack;
import java.io.*;

import quality.background;// or import quality.*;

class entrance
{
	public static void main(String args[])throws IOException
	{
		background b = new background();
		b.testing();
	}
}
```

Output



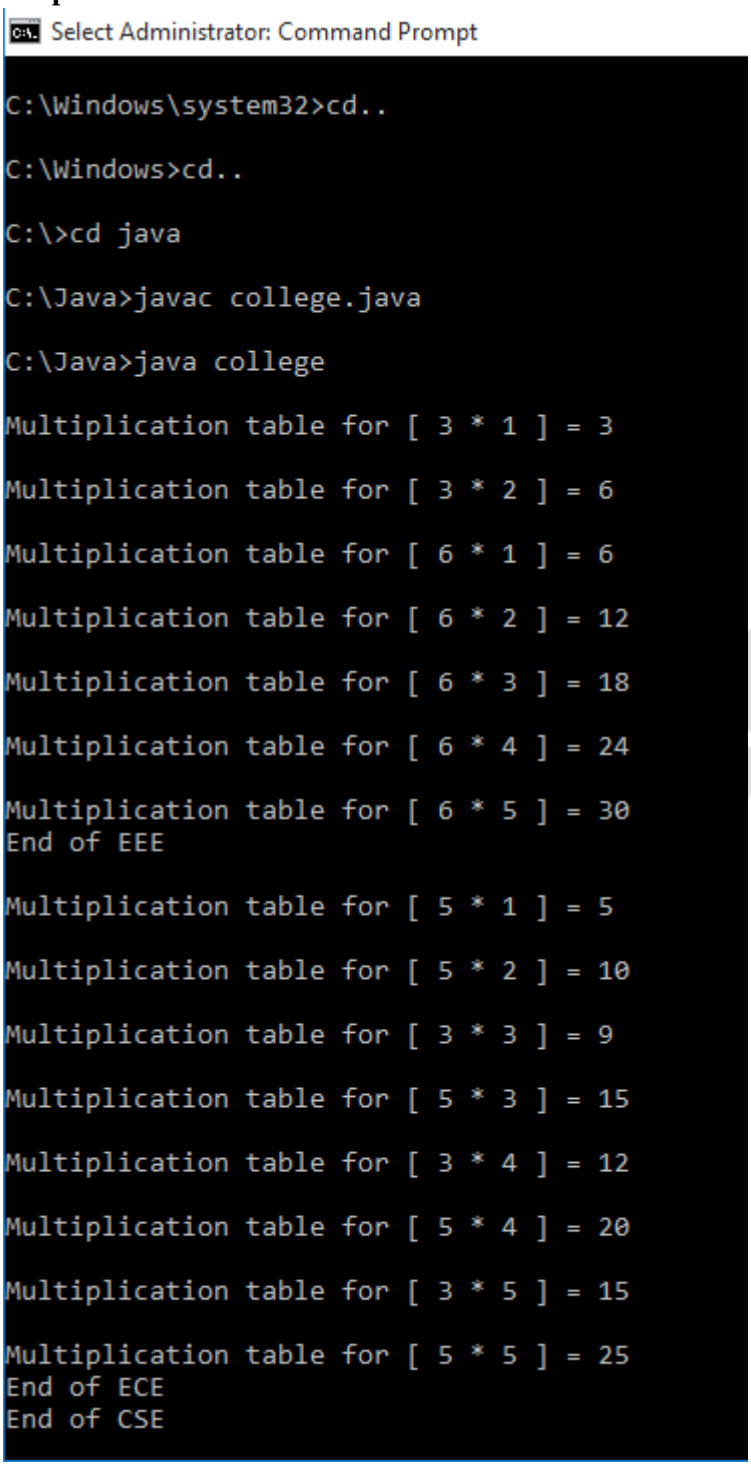
The following package named quality with the class named background shall be exported to the class named entrance that is contained in the package named mypack. .i.e. The following class named entrance that is contained in the package named mypack imports the class named background from the package named quality.

//Experiment 17
//Multithreading in Java

```
import java.io.*;
import java.lang.*;
import java.util.*;

class CSE extends Thread
{
	public void run()
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println("\nMultiplication table for [ "+3+" * "+i+" ] = "+(3*i));
		}
		System.out.println("End of CSE");
	}
}

class EEE extends Thread
{
	public void run()
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println("\nMultiplication table for [ "+6+" * "+i+" ] = "+(6*i));
		}
		System.out.println("End of EEE");
	}
}

class ECE extends Thread
{
	public void run()
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println("\nMultiplication table for [ "+5+" * "+i+" ] = "+(5*i));
		}
		System.out.println("End of ECE");
	}
}

class college
{
```

```
        public static void main(String args[])
        {
                new CSE().start();
                new EEE().start();
                new ECE().start();

        }
}
```

Output


Select Administrator: Command Prompt
C:\Windows\system32>cd..
C:\Windows>cd..
C:\>cd java
C:\Java>javac college.java
C:\Java>java college
Multiplication table for [3 * 1] = 3
Multiplication table for [3 * 2] = 6
Multiplication table for [6 * 1] = 6
Multiplication table for [6 * 2] = 12
Multiplication table for [6 * 3] = 18
Multiplication table for [6 * 4] = 24
Multiplication table for [6 * 5] = 30
End of EEE
Multiplication table for [5 * 1] = 5
Multiplication table for [5 * 2] = 10
Multiplication table for [3 * 3] = 9
Multiplication table for [5 * 3] = 15
Multiplication table for [3 * 4] = 12
Multiplication table for [5 * 4] = 20
Multiplication table for [3 * 5] = 15
Multiplication table for [5 * 5] = 25
End of ECE
End of CSE

//Experiment 18

//User defined exception with finally block

```
import java.io.*;
import java.lang.*;

class MyException extends Exception
{
	MyException(String message)
	{
		super(message);
	}
}

class test
{
	public static void main(String args [])
	{
		int x=5,y=1000;
		try
		{
			float  z = (x/y);
			if(z<0.01)
			{
				throw new MyException("\nNumber is too small");
			}
		}

		catch (MyException e)
		{
			System.out.println("\nCaught my exception");
			System.out.println(e);
		}

		finally
		{
			System.out.println("Prove the error");
		}
	}
}
```

Output