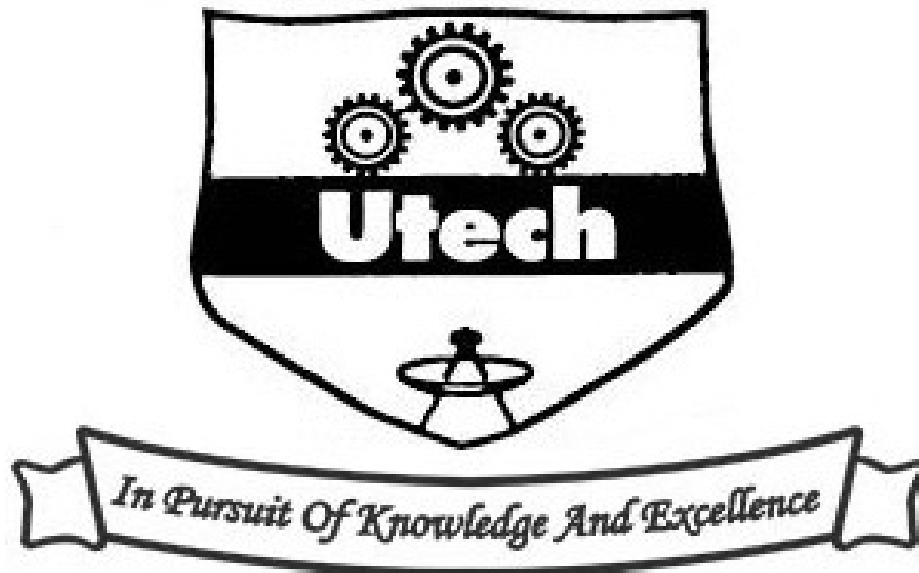


# Maulana Abul Kalam Azad University of Technology

(Java Assignment)

MAULANA ABUL KALAM AZAD  
UNIVERSITY OF TECHNOLOGY,  
WEST BENGAL



**Rohit Das**

B. Tech(CSE),5th Year

Roll No.: 30000114022

Regn. No.: 143000110023

Taught by: Prof. Kaushik Majumdar/ Prof. Debashis De

---

## INDEX

Sl. No.	Particulars	Date	Page No.	Signature
1.	Program to calculate average of cricket players.	28/9/16	3	
2.	Program to calculate cost for consumer.	28/9/16	4	
3.	Program to display votes received in election.	28/9/16	5	
4.	Program for operation on array elements.	19/10/16	6	
5.	Program to calculate factorial of a number.	19/10/16	7	
6.	Program to calculate sum of series $1+22+32+42+....$	19/10/16	7	
7.	Program to add values of Complex no. objects.	9/11/16	8	
8.	Program to convert distances to inches and add.	28/9/16	9	
9.	Program for String class with different constructors.	9/11/16	10	
10.	Program to display unique roll of Student objects.	9/11/16	10	
11.	Program to swap values of objects using 'friend' function.	9/11/16	11	
12.	Program for operation on matrices.	16/11/16	12	
13.	Program to use overloaded operators on float numbers.	16/11/16	14	
14.	Program to compare two strings by '==' operator.	16/11/16	15	
15.	Program to implement current and savings a/c.	23/11/16	16	
16.	Previous program rewritten using constructors.	23/11/16	18	
17.	Program for an educational institution.	23/11/16	20	
18.	Program to implement a network.	23/11/16	23	
19.	Program to show overriding using 'virtual'.	30/11/16	24	
20.	Program to implement Shape class using 'virtual'.	30/11/16	25	
21.	Previous program rewritten without using 'virtual'.	30/11/16	26	
22.	Program to use template for an array.	7/12/16	27	
23.	Program to implement exception handling with multiple catch.	7/12/16	28	
24.	Program to implement rethrowing an exception.	7/12/16	29	
25.	Program to implement a generic vector.	27/12/16	29	

---

1. Write a program to print the patterns below:

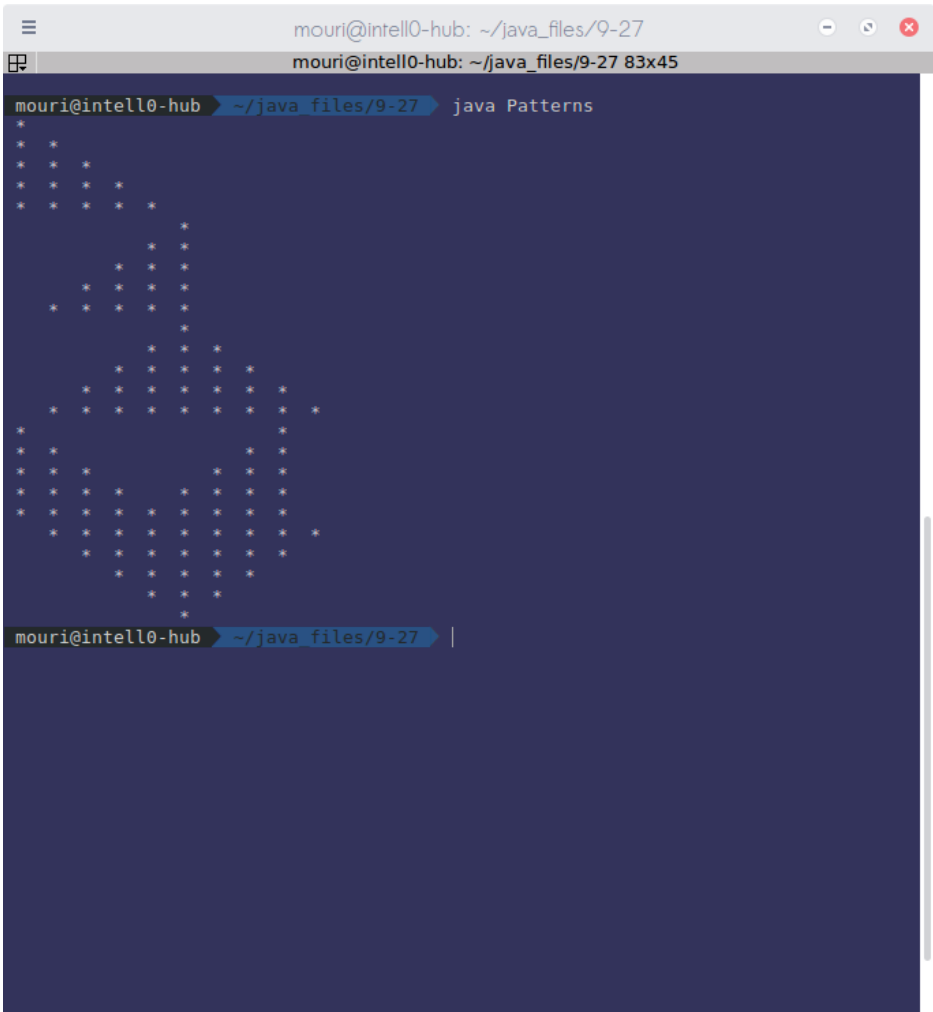
1. \* 2. \* 3.\* 4.\* 5.

Program:

```
class Patterns{
    void pattern1(int n){
        for(int i=1;i<=n;i++){
            int j=1;
            while(j<=i){
                System.out.print(" * ");
                j++;
            }
            System.out.println();
        }
    }
    void pattern2(int n){
        for(int i=1;i<=n;i++){
            for(int j=n-i;j>=0;j--)System.out.print(" ");
            for(int j=1;j<=i;j++)System.out.print(" * ");
            for(int j=i;j<i;j++)System.out.print(" * ");
            System.out.println();
        }
    }
    void pattern3(int n){
        for(int i=1;i<=n;i++){
            for(int j=n-i;j>=0;j--)System.out.print(" ");
            for(int j=1;j<=i;j++)System.out.print(" * ");
            for(int j=1;j<i;j++)System.out.print(" * ");
            System.out.println();
        }
    }
    void pattern4(int n){
        for(int i=n;i>0;i--){
            for(int j=n-i;j>=0;j--)System.out.print(" ");
            for(int j=1;j<=i;j++)System.out.print(" * ");
            for(int j=1;j<i;j++)System.out.print(" * ");
            System.out.println();
        }
    }
    void pattern5(int n){
        for(int i=n-1;i>0;i--){
            for(int j=n-i-1;j>=0;j--)System.out.print(" * ");
            for(int j=1;j<=i;j++)System.out.print(" ");
            for(int j=1;j<i;j++)System.out.print(" ");
            for(int j=n-i-1;j>=0;j--)System.out.print(" * ");
            System.out.println();
        }
        for(int j=1;j<=(2*n-1);j++)System.out.print(" * ");
        System.out.println();
    }
    public static void main(String [] args){
        Patterns obj=new Patterns();
        obj.pattern1(5);
        obj.pattern2(5);
    }
}
```

```
obj.pattern3(5);      obj.pattern5(5);
obj.pattern4(5);
}
```

Output:



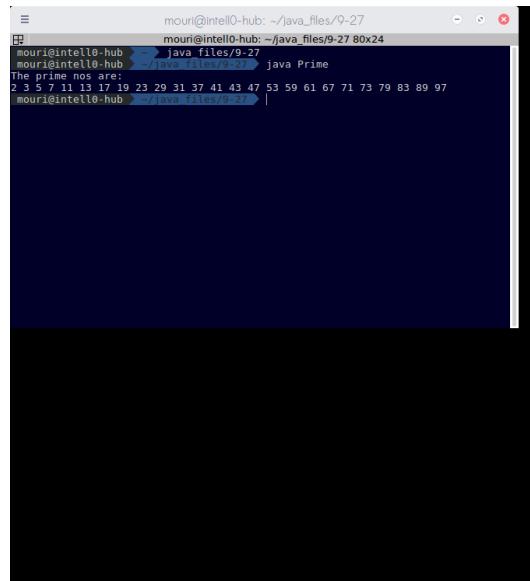
2. Write a program to print all prime numbers between 1 and 100.

Program:

```
class Prime{
    boolean isPrime(int n){
        int d=n/2;
        if(n==0 || n==1)return false;
        for(int i=2;i<=d;i++)    if(n%i==0)    return false;
        return true;
    }
    public static void main(String [] args){
        Prime obj=new Prime();
        System.out.println("The prime nos are:");
        for(int i=1;i<=100;i++)if(obj.isPrime(i))
            System.out.print(i+" ");
        System.out.println();
    }
}
```

```
}
```

Output:



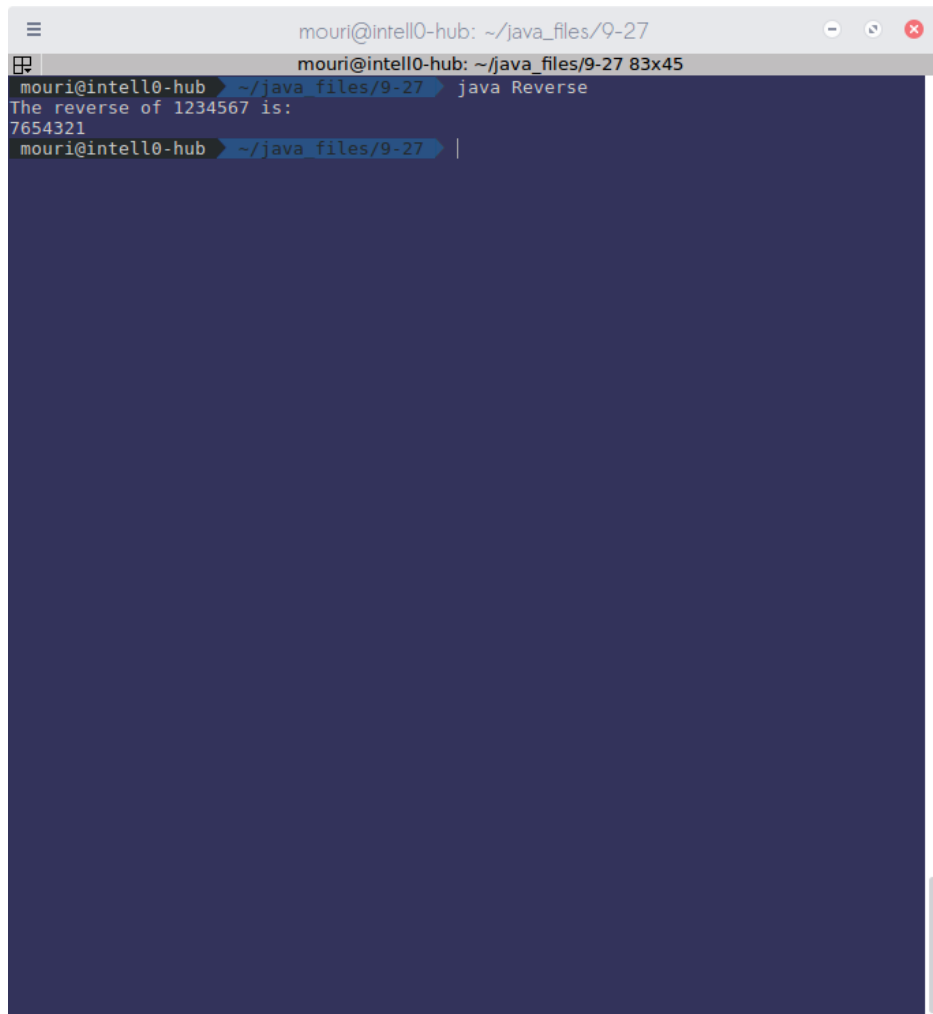
```
mour@intell0-hub: ~/java_files/9-27
mour@intell0-hub: ~/java_files/9-27$ java Prime
The prime nos are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

### 3. Write a program to print the reverse of a number.

Program:

```
class Reverse{
    int rev(int n){
        int s=0;
        while(n!=0){
            int r=n%10;
            s=s*10+r ;
            n/=10;
        }
        return s;
    }
    public static void main(String [] args){
        Reverse obj=new Reverse();
        System.out.println("The reverse of 1234567 is:");
        System.out.println(obj.rev(1234567));
    }
}
```

Output:

A screenshot of a terminal window with a dark blue background. The window title is "mouri@intell0-hub: ~/java\_files/9-27". The prompt is "mouri@intell0-hub: ~/java\_files/9-27 83x45". The user has entered the command "java Reverse". The output of the program is "The reverse of 1234567 is: 7654321". The prompt is now "mouri@intell0-hub: ~/java\_files/9-27 |".

```
mouri@intell0-hub: ~/java_files/9-27 83x45
mouri@intell0-hub: ~/java_files/9-27 java Reverse
The reverse of 1234567 is:
7654321
mouri@intell0-hub: ~/java_files/9-27 |
```

4. Write a program to print the sum of digits of a number.

Program:

```
class SumOfDigits{
    int sum(int n){
        int sum=0;
        while(n!=0){
            int r=n%10;
            sum+=r;
            n/=10;
        }
        return sum;
    }
    public static void main(String [] args){
        SumOfDigits obj=new SumOfDigits();
        System.out.println("The sum of digits for 1234567 is ::");
        System.out.println(obj.sum(1234567));
    }
}
```

Output:

```
mouri@intell0-hub: ~/java_files/9-27
mouri@intell0-hub: ~/java_files/9-27 83x45
mouri@intell0-hub ~/java_files/9-27 java SumOfDigits
The sum of digits for 1234567 is::
28
mouri@intell0-hub ~/java_files/9-27 |
```

5. Write a program to calculate the area of a circle, square, rectangle and triangle using function overloading.

Program:

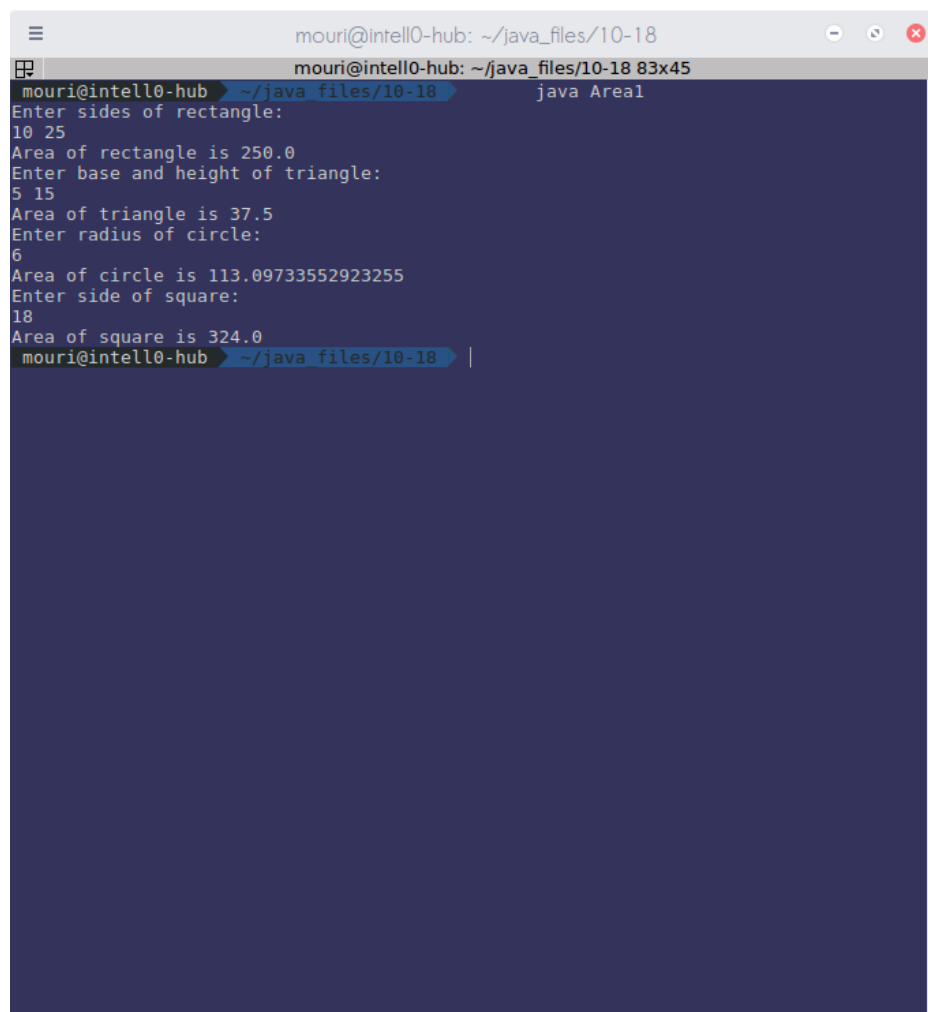
```
import java.util.*;
class Area1{
    private float area(float l,float b){
        return l*b;
    }
    private double area(double p,double base){
        return ((1/2.0)*base*p);
    }
    private double area(double r){
        return (Math.PI*r*r);
    }
    private float area(float a){
        return a*a;
    }
    public static void main(String [] args){
        Area1 obj=new Area1();
        Scanner sc=new Scanner(System.in);
```

```

        double p, base, r;
        float l, b, a;
        System.out.println("Enter sides of rectangle:");
        l=sc.nextFloat();
        b=sc.nextFloat();
        System.out.println("Area of rectangle is "+
        obj.area(l,b));
        System.out.println("Enter base and height of triangle:");
        p=sc.nextDouble();
        base=sc.nextDouble();
        System.out.println("Area of triangle is "+
        obj.area(p,base));
        System.out.println("Enter radius of circle:");
        r=sc.nextDouble();
        System.out.println("Area of circle is "+obj.area(r));
        System.out.println("Enter side of square:");
        a=sc.nextFloat();
        System.out.println("Area of square is "+obj.area(a));
    }
}

```

Output:



```

mouri@intell0-hub: ~/java_files/10-18
mouri@intell0-hub: ~/java_files/10-18 83x45
mouri@intell0-hub ~/java_files/10-18 java Areal
Enter sides of rectangle:
10 25
Area of rectangle is 250.0
Enter base and height of triangle:
5 15
Area of triangle is 37.5
Enter radius of circle:
6
Area of circle is 113.09733552923255
Enter side of square:
18
Area of square is 324.0
mouri@intell0-hub ~/java_files/10-18 |

```



---

## 6. Write the previous program using constructor overloading.

Program:

```
import java.util.*;
class Area2{
    static Scanner sc=new Scanner(System.in);
    static double p,base,r;
    static float l,b,a;
    Area2(float l,float b){ this.l=l; this.b=b; }
    Area2(float a){ this.a=a; }
    Area2(double p,double base){ this.p=p; this.base=base; }
    Area2(double r){ this.r=r; }
    private double area1(){ return l*b; }
    private double area2(){ return a*a; }
    private double area3(){ return (1/2.0)*base*p; }
    private double area4(){ return Math.PI*r*r; }
    public static void main(String [] args){
        System.out.println("Enter sides of rectangle:");
        l=sc.nextFloat(); b=sc.nextFloat();
        Area2 obj1=new Area2(l,b);
        System.out.println("Area of rectangle is "+obj1.area1());
        System.out.println("Enter sides of square:");
        a=sc.nextFloat();
        Area2 obj2=new Area2(a);
        System.out.println("Area of square is "+obj2.area2());
        System.out.println("Enter sides of triangle:");
        p=sc.nextDouble(); base=sc.nextDouble();
        Area2 obj3=new Area2(p,base);
        System.out.println("Area of triangle is "+obj3.area3());
        System.out.println("Enter radius of circle:");
        r=sc.nextDouble();
        Area2 obj4=new Area2(r);
        System.out.println("Area of circle is "+obj4.area4());
    }
}
```

Output:

```
mouri@intell0-hub: ~/java_files/10-18
mouri@intell0-hub: ~/java_files/10-18 83x45
mouri@intell0-hub ~/java_files/10-18 java Area2
Enter sides of rectangle:
10 15
Area of rectangle is 150.0
Enter sides of square:
18
Area of square is 324.0
Enter sides of triangle:
5 15
Area of triangle is 37.5
Enter radius of circle:
6
Area of circle is 113.09733552923255
mouri@intell0-hub ~/java_files/10-18 |
```

7. Write a program to design a class representing a bank account. The class should have the following data members:
- \* a/c no. \* customer id \* balance amount
- The class should have member methods with the following functions:
- \* initialize initial value
  - \* to deposit amount
  - \* to withdraw amount
  - \* to display customer id, a/c no. and current balance.

Program:

```
import java.util.*;
class Bank{
    static Scanner sc=new Scanner(System.in);
    static long acno;          static double amt;
    static String id;
    private void init(){ acno=0; amt=0.0; id=""; }
    private double deposit(double d){ return amt+=d; }
    private double withdraw(double d)
    {
        if(d<amt&&amt!=0)return amt-=d;
        else {
```

```

        System.out.println("Not Enough Balance!!");
        return amt;
    }
}
private void print(){
    System.out.println("Customer ID \t A/c No. \t Current
    Balance");
    System.out.println(id+"\t \t "+acno+"\t \t "+amt);
}
public static void main(String [] args){
    Bank obj=new Bank();
    obj.init();
    System.out.println("Enter account no and current balance
    :");
    id="3000114022";
    acno=sc.nextLong(); amt=sc.nextDouble();
    double d=0.0; int choice=0;
    do{
        System.out.println("Main Menu");
        System.out.println("0. Deposit");
        System.out.println("1. Withdrawal");
        System.out.println("2. Print Statement");
        System.out.println("3. Exit");
        System.out.println("Enter choice:");
        choice=sc.nextInt();
        switch(choice){
            case 0:d=0.0;
                System.out.println("Enter amount to
                deposit:");
                d=sc.nextDouble();
                System.out.println("The deposit is "+d+
                ". The current balance is "+
                (double)obj.deposit(d));
                break;
            case 1:d=0.0;
                System.out.println("Enter amount to
                withdraw:");
                d=sc.nextDouble();
                System.out.println("The withdrawal is
                "+d+". The current balance is "+
                (double)obj.withdraw(d));
                break;
            case 2:obj.print();
                break;
            default: System.out.println("Thank you");
                break;
        }
    }while(choice<3);
}
}

```

Output:

```
mouri@intell0-hub: ~/java_files/10-18
mouri@intell0-hub: ~/java_files/10-18 83x45
mouri@intell0-hub: ~/java_files/10-18 java Bank
Enter account no and current balance:
1200
10000
Main Menu
0. Deposit
1. Withdrawal
2. Print Statement
3. Exit
Enter choice:
0
Enter amount to deposit:
2000
The deposit is 2000.0. The current balance is 12000.0
Main Menu
0. Deposit
1. Withdrawal
2. Print Statement
3. Exit
Enter choice:
1
Enter amount to withdraw:
15000
Not Enough Balance!!
The withdrawal is 15000.0. The current balance is 12000.0
Main Menu
0. Deposit
1. Withdrawal
2. Print Statement
3. Exit
Enter choice:
1
Enter amount to withdraw:
3000
The withdrawal is 3000.0. The current balance is 9000.0
Main Menu
0. Deposit
1. Withdrawal
2. Print Statement
3. Exit
Enter choice:
3
Thank you
mouri@intell0-hub: ~/java_files/10-18
```

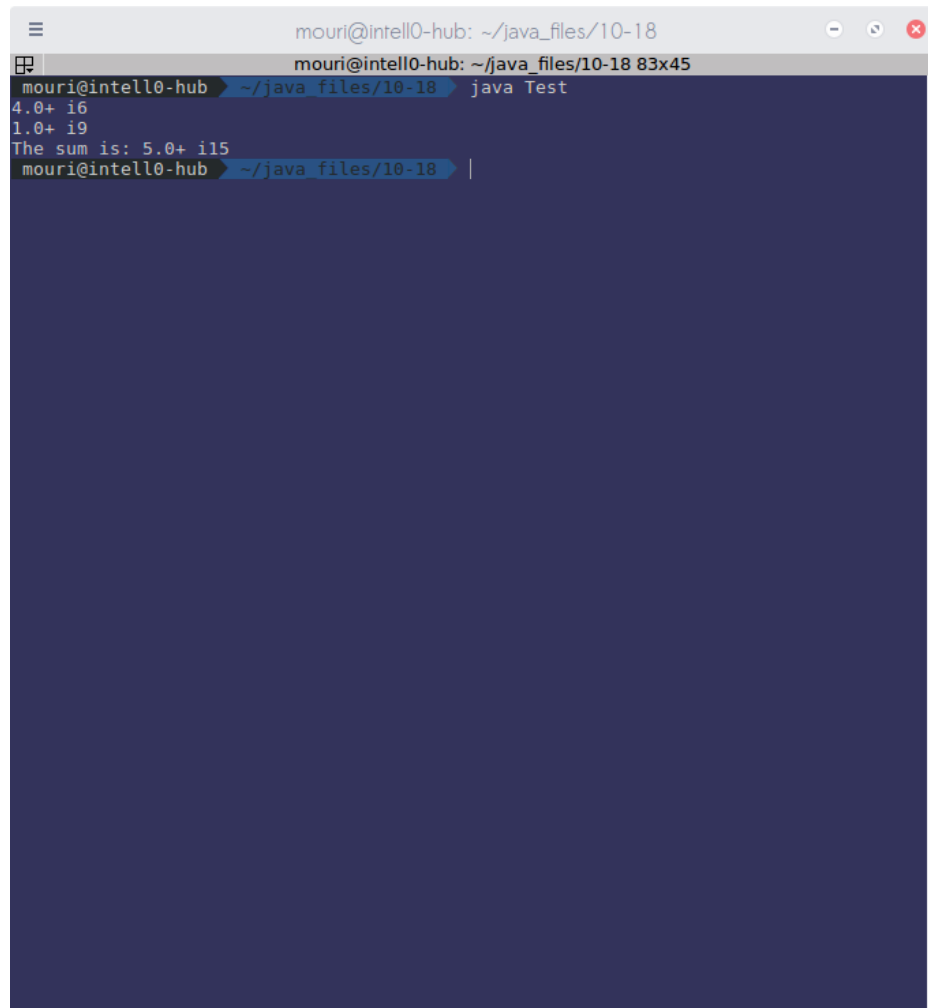
8. Write a program to add two complex numbers. Print the result in  $(x + iy)$  form. Use objects as arguments to a method which will perform the addition and use function overloading.

Program:

```
class Complex{
    double x;    int y;
    Complex(double a,int b){
        x=a;    y=b;
    }
    void print(){    System.out.println(x+" i"+y);    }
}
class Test
{
    double real;    int imag;
    private double sum(double a,double b){
        real=a+b;    return real;
    }
    private int sum(int a,int b){
        imag=a+b;    return imag;
    }
}
```

```
public static void main(String [] args){
    Complex obj=new Complex(4,6);
    obj.print();
    Complex obj1=new Complex(1,9);
    obj1.print();
    Test t1=new Test();
    System.out.println("The sum is: "+t1.sum(obj.x,obj1.x)+"+ i"+
    t1.sum(obj.y,obj1.y));
}
```

Output:

A screenshot of a terminal window with a dark blue background. The window title is 'mouri@intell0-hub: ~/java\_files/10-18'. The prompt is 'mouri@intell0-hub: ~/java\_files/10-18 83x45'. The command 'java Test' has been executed. The output is: '4.0+ i6', '1.0+ i9', and 'The sum is: 5.0+ i15'. The prompt is now 'mouri@intell0-hub: ~/java\_files/10-18 |' with a cursor.

```
mouri@intell0-hub: ~/java_files/10-18
mouri@intell0-hub: ~/java_files/10-18 83x45
mouri@intell0-hub: ~/java_files/10-18 java Test
4.0+ i6
1.0+ i9
The sum is: 5.0+ i15
mouri@intell0-hub: ~/java_files/10-18 |
```

9. Write a program to design a class Student. The class should have the following data members:

\* name \* roll \* sub-marks \* sports-marks \* grades.

The class should have the following functions:

\* initialize values for data members,

\* calculate total marks,

\* display total marks

\* display grade

Program:

```
import java.util.*;
class Student{
    private static String name;
    private static long roll;
    private static int submarks,sportsmarks;
    private void init(){
        roll=submarks=sportsmarks=0;
        name="";
    }
    private long total(){    return submarks+sportsmarks;    }
    private void print(){
        System.out.println("Subject Marks: "+submarks+" out of 50.");
        System.out.println("Sports Marks: "+sportsmarks+" out of 50.");
        System.out.println("Total Marks: "+(submarks+sportsmarks)+" out of 100.");
    }
    private char grade(){
        long total=submarks+sportsmarks;
        if(total>90)return 'A';
        else if(total<=90 && total >=80)return 'B';
        else if(total<80 && total >=70)return 'C';
        else return 'D';
    }
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        Student obj=new Student();  obj.init();
        System.out.println("Enter name of student:");
        name=sc.nextLine();
        System.out.println("Enter roll no., subject marks(out of 50) and sports marks(out of 50):");
        roll=sc.nextLong();    submarks=sc.nextInt();
        sportsmarks=sc.nextInt();
        System.out.println("The total is "+obj.total());
        System.out.println("The grade of the student is: "+obj.grade());
    }
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-9 ./a.out
Hello World
```

10. Write a program to create a student class and create 10 objects and display the roll no. of the students. Use static data member and array to create objects. Do not use constructors.

Program:

```
#include<iostream>
#include<cstdio>
using namespace std;
class Student{
    int roll;
    public:
        static int count;
        Student(){}
        void init(){    roll=++count;        }
        void display(){ printf("%d ",roll);    }
};
int Student::count;
int main(){
    Student arr[10];
    for(int i=0;i<10;i++){
        int x=Student::count;
        arr[i].init();
    }
    printf("The rolls are: \n");
    for(int i=0;i<10;i++){
        printf("Student %d: ",i);
        arr[i].display();
        printf("\n");
    }
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-9 ./a.out
The rolls are:
Student 0: 1
Student 1: 2
Student 2: 3
Student 3: 4
Student 4: 5
Student 5: 6
Student 6: 7
Student 7: 8
Student 8: 9
Student 9: 10
```

11. Write a program to values of objects using 'friend' functions.

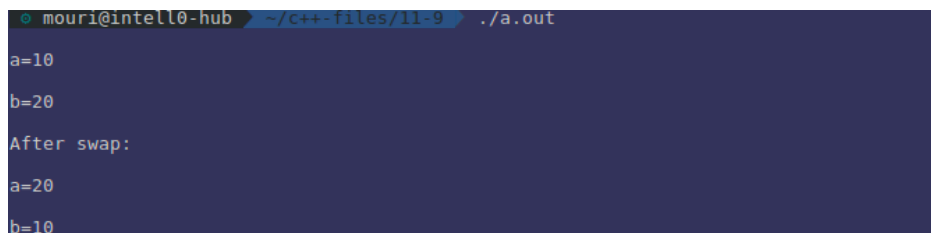
Program:

```

#include<iostream>
#include<cstdio>
using namespace std;
class Swap
{
    int n;
    public:
        Swap(){}
        Swap(int n){    this->n=n;    }
        void show(){    printf("%d \n",n);    }
        friend void swap(Swap,Swap);
};
class Swap2{
    public:
        friend void swap(Swap a,Swap b){
            Swap temp;
            printf("\na="); a.show();
            printf("\nb="); b.show();
            printf("\nAfter swap:\n");
            temp.n=a.n;  a.n=b.n;  b.n=temp.n;
            printf("\na="); a.show();
            printf("\nb="); b.show();
        }
};
int main(){
    Swap a(10);    Swap b(20);
    Swap2 s;    swap(a,b);
}

```

Output:



```

mouri@intel10-hub ~/c++-files/11-9 ./a.out
a=10
b=20
After swap:
a=20
b=10

```

## 12. Write a program to create a class Matrix of size m x n. Define all possible matrix operations for Matrix type objects.

Program:

```

#include<iostream>
#include<cstdio>
using namespace std;
class Matrix{
    int r,c;
    public:
    int a[100][100];

```



```

Matrix(){}
Matrix(int row,int col){          r=row;c=col;      }
Matrix operator+(Matrix);
Matrix operator-(Matrix);
Matrix operator*(Matrix);
void input(void);
void display(void);
};
Matrix Matrix::operator+(Matrix c){
    Matrix temp(2,2);
    for(int i=0;i<c.r;i++){
        for(int j=0;j<c.c;j++){
            temp.a[i][j]=a[i][j]+c.a[i][j];
        }
    }
    return (temp);
}
Matrix Matrix::operator-(Matrix c){
    Matrix temp(2,2);
    for(int i=0;i<c.r;i++){
        for(int j=0;j<c.c;j++){
            temp.a[i][j]=a[i][j]-c.a[i][j];
        }
    }
    return (temp);
}
Matrix Matrix::operator*(Matrix c){
    Matrix temp(2,2);
    for(int i=0;i<r;i++){
        for(int j=0;j<c.c;j++){
            for(int k=0;k<c.r;k++){
                temp.a[i][j]=a[i][k]*c.a[k][j];
            }
        }
    }
    return (temp);
}
void Matrix::input(void){
    printf("Enter data for matrix(%d x %d):",r,c);
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            scanf("%d",&a[i][j]);
        }
    }
}
void Matrix::display(void){
    printf("The data for matrix \n");
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++)printf("%d ",a[i][j]);
        printf("\n");
    }
}

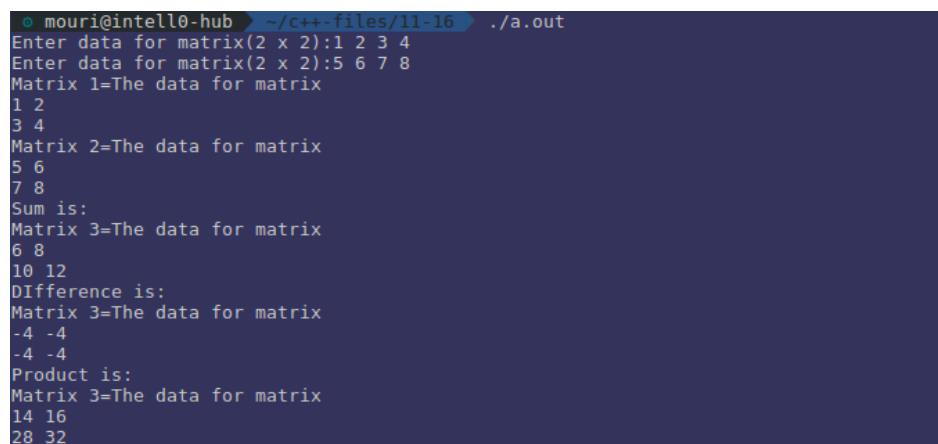
```

```

int main(){
    Matrix m1,m2,m3;
    m1=Matrix(2,2); m1.input();
    m2=Matrix(2,2); m2.input();
    m3=m1+m2;
    printf("Matrix 1=");    m1.display();
    printf("Matrix 2=");    m2.display();
    cout<<"Sum is:"<<endl;
    printf("Matrix 3=");    m3.display();
    m3=m1-m2;
    cout<<"Difference is:"<<endl;
    printf("Matrix 3=");    m3.display();
    m3=m1*m2;
    cout<<"Product is:"<<endl;
    printf("Matrix 3=");    m3.display();
    return 0;
}

```

Output:



```

mouri@intell0-hub ~/c++-files/11-16 ./a.out
Enter data for matrix(2 x 2):1 2 3 4
Enter data for matrix(2 x 2):5 6 7 8
Matrix 1=The data for matrix
1 2
3 4
Matrix 2=The data for matrix
5 6
7 8
Sum is:
Matrix 3=The data for matrix
6 8
10 12
Difference is:
Matrix 3=The data for matrix
-4 -4
-4 -4
Product is:
Matrix 3=The data for matrix
14 16
28 32

```

13. Write a program to create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.

Program:

```

#include<iostream>
#include<cstdio>
using namespace std;
class Float{
    float x;
public:
    Float(){}
    Float(float a){ x=a; }
    Float operator+(Float);
    Float operator-(Float);
    Float operator*(Float);
    Float operator/(Float);
}

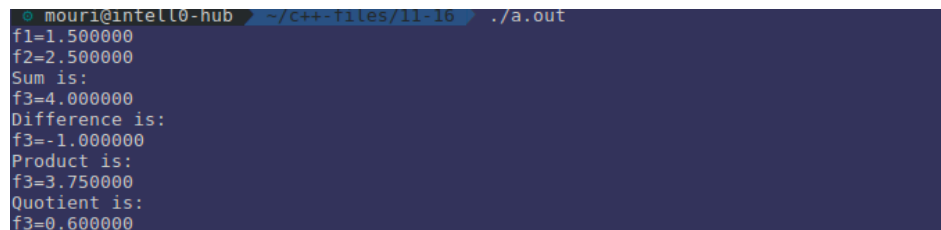
```

```

        void display(void);
};
Float Float::operator+(Float c){          return Float(x+c.x);      }
Float Float::operator-(Float c){          return Float(x-c.x);      }
Float Float::operator*(Float c){          return Float(x*c.x);      }
Float Float::operator/(Float c){          return Float(x/c.x);      }
void Float::display(void){                printf("%f \n",x);        }
int main(){
    Float f1,f2,f3;
    f1=Float(1.5);  f2=Float(2.5);
    f3=f1+f2;
    printf("f1=");  f1.display();
    printf("f2=");  f2.display();
    cout<<"Sum is:"<<endl;
    printf("f3=");  f3.display();
    cout<<"Difference is:"<<endl;
    f3=f1-f2;
    printf("f3=");  f3.display();
    cout<<"Product is:"<<endl;
    f3=f1*f2;
    printf("f3=");  f3.display();
    cout<<"Quotient is:"<<endl;
    f3=f1/f2;
    printf("f3=");  f3.display();
    return 0;
}

```

Output:



```

mouri@intel0-hub ~/c++-files/11-16 ./a.out
f1=1.500000
f2=2.500000
Sum is:
f3=4.000000
Difference is:
f3=-1.000000
Product is:
f3=3.750000
Quotient is:
f3=0.600000

```

#### 14. Write a program to compare two strings by overloading the '==' operator.

Program:

```

#include<iostream>
#include<cstdio>
#include<string>
using namespace std;
class String{
    string s;
public:
    String(){          s="";      }
    String(string _s){          s=_s;      }
    string operator==(String);

```

```
};
string String::operator==(String str){
    if(s==str.s) return "Equals\n";
    else return "Not equals\n";
}
int main(){
    String s1("Hello"),s2("Hola"),s3("Hello");
    cout<<"Comparing Hello and Hola:"<<endl;
    cout<<(s1==s2);
    cout<<"Comparing Hello and Hello:"<<endl;
    cout<<(s1==s3);
}
```

Output:

```
mouri@intell0-hub: ~/C++-files/11-16 $ ./a.out
Comparing Hello and Hola:
Not equals
Comparing Hello and Hello:
Equals
```

15. Write a program to create a class account that stores customer name, account number and type of account. Create two more classes for current a/c and savings a/c. The current a/c will have:
  - \* cheque facility,
  - \* minimum balance and deduction for balance below that,
  - \* deposit and withdrawal. The saving a/c will have similar member methods, except for cheque and minimum balance, it will have an interest calculation. Do not use constructors.

Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Account{
    string customer_name, acct_type;
    ll acct;
};
class Cur_acct:public Account{
    double balance, minBal;
public:
    void setbal(double a, double b) {
        balance=a;
        minBal=b;
    }
    void withdrawal(int n) {
        if(balance<minBal) balance-=(balance*0.10);
        else if(n>balance) printf("Insufficient balance.\n");
    }
};
```

```

        else balance-=n;
    }
    void deposit(int n){    balance+=n;    }
    void cheque(){ cout<<"Deposited by cheque..."<<endl;    }
    void show(){
        printf("Current balance=%.2f\n", balance);
    }
};

class Sav_acct:public Account{
    double balance;
public:
    void setbal(double a){    balance=a;    }
    void withdrawal(int n){
        if(n>balance)printf("Insufficient balance.\n");
        else balance-=n;
    }
    void deposit(int n){    balance+=n;    }
    void interest(){    balance+=(balance*0.20);    }
    void show(){
        printf("Current balance: %.2f\n", balance);
    }
};

int main(){
    cout<<"Enter type of account(s for savings, c for current):"<<endl;
    char c;    cin>>c;
    cout<<"Enter balance:"<<endl;
    int bal;    cin>>bal;
    if(c=='c'){
        cout<<"Enter minimum balance:"<<endl;
        int minbal;    cin>>minbal;
        Cur_acct c1;
        c1.setbal(bal,minbal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;    cin>>n;
        cout<<"Do you want to use cheque or cash?(c for cheque)..
"<<endl;
        char c;    cin>>c;
        if(c=='c')c1.cheque();
        c1.deposit(n);    cout<<"Deposited: Rs."<<n<<"\n";
        c1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n;    c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";    c1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n;    c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";    c1.show();
    }
    else{
        Sav_acct s1;
        s1.setbal(bal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;    cin>>n;
    }
}

```

```

        s1.deposit(n);
        cout<<" Deposited: Rs."<<n<<"\n";          s1.show();
        cout<<" Enter amount to withdraw:"<<endl;
cin>>n; s1.withdrawal(n);
        cout<<" Withdrawal: Rs."<<n<<"\n";
        s1.show();
        cout<<" Enter amount to withdraw:"<<endl;
cin>>n; s1.withdrawal(n);
        cout<<" Withdrawal: Rs."<<n<<"\n";          s1.show();
        cout<<" Adding interest of 0.20 to current balance:"
<<endl;
        s1.interest();  s1.show();
    }
    return 0;
}

```

Output:

```

mouri@intell0-hub: ~/c++-files/11-23
mouri@intell0-hub: ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
s
Enter balance:
12000
Enter amount to deposit:
2000
Deposited: Rs.2000
Current balance:14000.00
Enter amount to withdraw:
20000
Insufficient balance.
Withdrawal: Rs.20000
Current balance:14000.00
Enter amount to withdraw:
5000
Withdrawal: Rs.5000
Current balance:9000.00
Adding interest of 0.20 to current balance:
Current balance:10800.00

mouri@intell0-hub: ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
c
Enter balance:
12000
Enter minimum balance:
2000
Enter amount to deposit:
2000
Do you want to use cheque or cash?(c for cheque)..
c
Deposited by cheque...
Deposited: Rs.2000
Current balance=14000.00
Enter amount to withdraw:
13000
Withdrawal: Rs.13000
Current balance=1000.00
Enter amount to withdraw:
2000
Withdrawal: Rs.2000
Current balance=900.00

```

## 16. Rewrite the above program using constructors.

Program:

```

#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
using namespace std;
class Account{
    string customer_name, acct_type;
    ll acct;
public:
    Account(){}
    Account(string a,string b,ll c){

```

```

        customer_name=a;
        acct_type=b;
        acct=c;
    }
};
class Cur_acct:public Account{
    double balance,minBal;
public:
    Cur_acct(double a,double b){
        balance=a;
        minBal=b;
    }
    void withdrawal(int n){
        if(balance<minBal)balance-=(balance*0.10);
        else if(n>balance)printf("Insufficient balance.\n");
        else balance-=n;
    }
    void deposit(int n){    balance+=n;    }
    void cheque(){
        cout<<"Deposited by cheque..."<<endl;
    }
    void show(){
        printf("Current balance=%.2f\n",balance);
    }
};
class Sav_acct:public Account
{
    double balance;
public:
    Sav_acct(int a){    balance=a;    }
    void withdrawal(int n){
        if(n>balance)printf("Insufficient balance.\n");
        else balance-=n;
    }
    void deposit(int n){    balance+=n;    }
    void interest(){    balance+=(balance*0.20);    }
    void show(){
        printf("Current balance:%.2f\n",balance);
    }
};
int main(){
    cout<<"Enter type of account(s for savings, c for current):"<<endl;
    char c;    cin>>c;
    cout<<"Enter balance:"<<endl;
    int bal;    cin>>bal;
    if(c=='c'){
        cout<<"Enter minimum balance:"<<endl;
        int minbal;    cin>>minbal;
        Cur_acct c1(bal,minbal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;    cin>>n;
        cout<<"Do you want to use cheque or cash?(c for cheque).."<<endl;
    }
}

```

```

        char c; cin>>c;
        if(c=='c')c1.cheque();
        c1.deposit(n);
        cout<<"Deposited: Rs."<<n<<"\n";          c1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n; c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";
        c1.show();
        cout<<"Enter amount to withdraw:"<<endl;          cin>>n;
        c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";          c1.show();
    }
    else{
        Sav_acct s1(bal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;  cin>>n;
        s1.deposit(n);
        cout<<"Deposited: Rs."<<n<<"\n";          s1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n;      s1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";
        s1.show();
        cout<<"Enter amount to withdraw:"<<endl;          cin>>n;
        s1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";          s1.show();
        cout<<"Adding interest of 0.20 to current balance:"<<endl;
        s1.interest();  s1.show();
    }
    return 0;
}

```

Output:

```

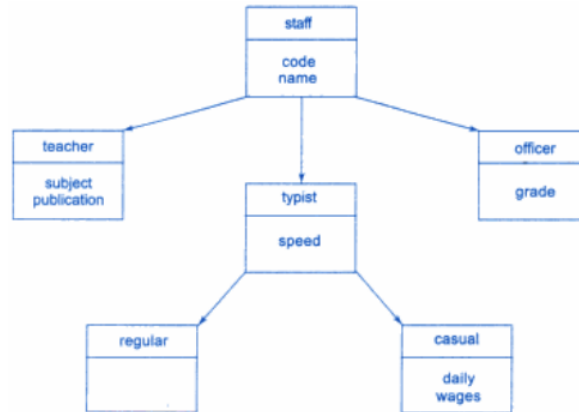
mouri@intell0-hub: ~/c++-files/
❖ mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
s
Enter balance:
12000
Enter amount to deposit:
2000
Deposited: Rs.2000
Current balance:14000.00
Enter amount to withdraw:
20000
Insufficient balance.
Withdrawal: Rs.20000
Current balance:14000.00
Enter amount to withdraw:
5000
Withdrawal: Rs.5000
Current balance:9000.00
Adding interest of 0.20 to current balance:
Current balance:10800.00

mouri@intell0-hub: ~/c++-files/1
❖ mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
c
Enter balance:
12000
Enter minimum balance:
2000
Enter amount to deposit:
2000
Do you want to use cheque or cash?(c for cheque)..
c
Deposited by cheque...
Deposited: Rs.2000
Current balance=14000.00
Enter amount to withdraw:
13000
Withdrawal: Rs.13000
Current balance=1000.00
Enter amount to withdraw:
2000
Withdrawal: Rs.2000
Current balance=900.00

```



17. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in figure below. The figure also shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required.



Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
using namespace std;
class Account{
    string customer_name , acct_type ;
    ll acct ;
public:
    Account(){}
    Account(string a , string b , ll c){
        customer_name=a;
        acct_type=b;
        acct=c;
    }
};
class Cur_acct:public Account{
    double balance , minBal;
public:
    Cur_acct(double a , double b){
        balance=a;
        minBal=b;
    }
    void withdrawal(int n){
        if ( balance<minBal) balance -= (balance * 0.10);
        else if (n>balance) printf(" Insufficient balance.\n");
        else balance-=n;
```

```

    }
    void deposit(int n){    balance+=n;    }
    void cheque(){
        cout<<"Deposited by cheque..."<<endl;
    }
    void show(){
        printf("Current balance=%.2f\n",balance);
    }
};
class Sav_acct:public Account
{
    double balance;
public:
    Sav_acct(int a){    balance=a;    }
    void withdrawal(int n){
        if(n>balance)printf("Insufficient balance.\n");
        else balance-=n;
    }
    void deposit(int n){    balance+=n;    }
    void interest(){    balance+=(balance*0.20);    }
    void show(){
        printf("Current balance: %.2f\n",balance);
    }
};
int main(){
    cout<<"Enter type of account(s for savings, c for current):"<<endl;
    char c;    cin>>c;
    cout<<"Enter balance:"<<endl;
    int bal;    cin>>bal;
    if(c=='c'){
        cout<<"Enter minimum balance:"<<endl;
        int minbal;    cin>>minbal;
        Cur_acct c1(bal,minbal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;    cin>>n;
        cout<<"Do you want to use cheque or cash?(c for cheque).."<<endl;
        char c;    cin>>c;
        if(c=='c')c1.cheque();
        c1.deposit(n);
        cout<<"Deposited: Rs."<<n<<"\n";    c1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n;    c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";
        c1.show();
        cout<<"Enter amount to withdraw:"<<endl;    cin>>n;
        c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";    c1.show();
    }
    else{
        Sav_acct s1(bal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;    cin>>n;
    }
}

```

```

        s1.deposit(n);
        cout<<" Deposited: Rs."<<n<<"\n";          s1.show();
        cout<<" Enter amount to withdraw:"<<endl;
cin>>n;      s1.withdrawal(n);
        cout<<" Withdrawal: Rs."<<n<<"\n";
        s1.show();
        cout<<" Enter amount to withdraw:"<<endl;      cin>>n;
        s1.withdrawal(n);
        cout<<" Withdrawal: Rs."<<n<<"\n";          s1.show();
        cout<<" Adding interest of 0.20 to current balance:"<<endl;
        s1.interest();  s1.show();
    }
    return 0;
}

```

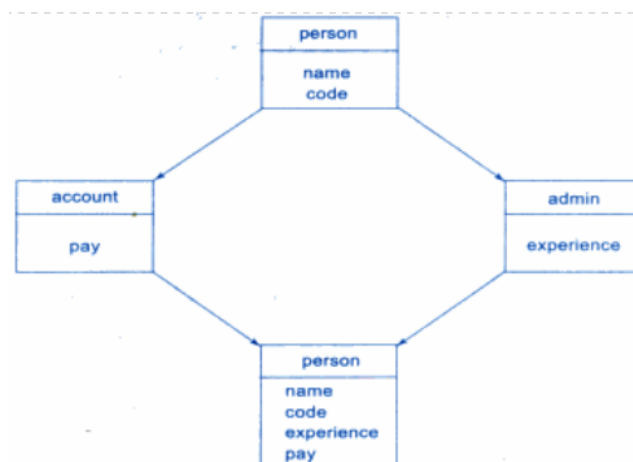
Output:

```

mouri@intell0-hub: ~/c++-files/11-23 83x45
❖ mouri@intell0-hub ~/c++-files/11-23 ./a.out
Staff: Teacher
Code= 0001
Name= A
Subject= Maths
Publication= Srijan
Staff: Officer
Code= 1001
Name= B
Grade= A
Staff: Typist(Regular)
Code= 2001
Name= C
Speed= 30 Words per minute.
Staff: Typist(Casual)
Code= 2002
Name= D
Speed= 30 Words per minute.
Daily Wages= 10000

```

18. Consider a class network as shown below. Define all four classes and write a program to create, update and display the information contained in master objects.



Program:

```
#include <iostream>
```

```

#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Person{
    public:
    string name,code;
    void init(string a,string b){
        name=a; code=b;
    }
    virtual void display()=0;
};
class Account:virtual public Person{
    public:
    int pay;
    void init1(string a,string b,int c){
        init(a,b);      pay=c;
    }
    void display(){
        cout<<"Name= "<<name<<endl;      cout<<"Code= "<<code<<endl;
        cout<<"Pay= "<<pay<<endl;
    }
};
class Admin:virtual public Person{
    public:
    string exp;
    void init2(string a,string b,string c){
        init(a,b);      exp=c;
    }
    void display(){
        cout<<"Name= "<<name<<endl;      cout<<"Code= "<<code<<endl;
        cout<<"Experience= "<<exp<<endl;
    }
};
class Master:public Account,public Admin{
    public:
    void init3(string a,string b,string c,int d){
        init1(a,b,d);      init2(a,b,c);
    }
    void display(){
        cout<<"Name= "<<name<<endl;      cout<<"Code= "<<code<<endl;
        cout<<"Experience= "<<exp<<endl;      cout<<"Pay= "<<pay<<endl;
    }
};
int main(){
    Master p;
    p.init3("A","0001","2 years",10000);      p.display();
    return 0;
}

```

Output:

```
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Name= A
Code= 0001
Experience= 2 years
Pay= 10000
```

## 19. Write a program demonstrating overriding using 'virtual'.

Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Base{
public:
    void display(){ cout<<"Display Base"<<endl; }
    virtual void show(){ cout<<"Show Base"<<endl; }
};
class Derived:public Base{
public:
    void display(){ cout<<"Display Derived"<<endl; }
    void show(){ cout<<"Show Derived"<<endl; }
};
int main(){
    Base b;
    Derived d;    Base *bptr;
    bptr=&b;
    bptr->display ();    bptr->show ();
    bptr=&d;
    bptr->display ();    bptr->show ();
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-30 ./a.out
Display Base
Show Base
Display Base
Show Derived
```

## 20. Write a program implementing Shape class, from which Triangle and Rectangle inherit. Use 'virtual'

Program:

```
#include<iostream>
#include<cstdio>
```

```

#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Shape{
public:
double l,b;
void getData(double _l ,double _b){
    l=_l;    b=_b;
}
virtual void display(){
    cout<<"Area ="<<(l*b)<<"\n" ;
}
};
class Triangle:public Shape{
public:
void display(){
    cout<<"Area of triangle="<<(0.5*b*l)<<"\n" ;
}
};
class Rectangle:public Shape{
public:
void display(){
    cout<<"Area of triangle="<<(l*b)<<"\n" ;
}
};
class Circle:public Shape{
public:
void getData(int _l ,int _b=0){
    l=_l;    b=_b;
}
void display(){
    cout<<"Area of circle="<<(3.1412*l*l)<<"\n" ;
}
};
int main()
{
    Triangle t; Rectangle r;
    Circle c;
    double l,b;
    t.getData(4,5);    t.display();
    r.getData(4,5);    r.display();
    c.getData(5);    c.display();
    return 0;
}

```

Output:

```

mouri@intell0-hub ~/c++-files/11-30 ./a.out
Area of triangle=10
Area of rectangle=20
Area of circle=78.53

```

---

## 21. Rewrite the previous program without using 'virtual'.

Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Shape{
    public:
    double l,b;
    void getData(double _l,double _b){
        l=_l;    b=_b;
    }
    void display(){
        cout<<"Area ="<<(l*b)<<"\n";
    }
};
class Triangle:public Shape{
    public:
    void display(){
        cout<<"Area of triangle="<<(0.5*b*l)<<"\n";
    }
};
class Rectangle:public Shape{
    public:
    void display(){
        cout<<"Area of rectangle="<<(l*b)<<"\n";
    }
};
class Circle:public Shape{
    public:
    void getData(int _l,int _b=0){
        l=_l;    b=_b;
    }
};
int main()
{
    Triangle t; Rectangle r;
    Circle c;
    t.getData(4,5);    t.display();
    r.getData(4,5);    r.display();
    c.getData(5);      c.display();
    return 0;
}
```

Output:

A terminal window with a dark blue background. The prompt is 'mouri@intel10-hub' followed by the path '~/c++-files/11-30' and the command './a.out'. The output consists of three lines: 'Area of triangle=10', 'Area of rectangle=20', and 'Area =0'.

```
mouri@intel10-hub ~/c++-files/11-30 ./a.out
Area of triangle=10
Area of rectangle=20
Area =0
```

---

## 22. Write a function template for finding the minimum value contained in an array.

Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
template<class R>
R minimum(R a[],int n){
    R m;    m=a[0];
    for(int i=1;i<3;i++)if(m>a[i])m=a[i];
    return m;
}
int main(){
    int x[3]={10,21,3};
    cout<<minimum(x,3)<<"\n";
    return 0;
}
```

Output:



```
mouri@intell0-hub ~/c++-files/12-7 ./a.out
3
```

## 23. Write a program containing a possible exception. Perform exception handling with multiple catch.

Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
void excp(){
    int x=4;    int y=4;
    if(x==y)throw(x-y);
    else cout<<"Its fine"<<endl;
}
int main(){
    try{        excp();    }
    catch(int i){
        cout<<"Both are equal."<<endl;
    }
}
```



```

    catch(char c){
        cout<<"Character is found."<<endl;
    }
    return 0;
}

```

Output:

```

mouri@intell0-hub ~/c++-files/12-7 ./a.out
Both are equal.

```

## 24. Write a program to demonstrate the concept of rethrowing an exception.

Program:

```

#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
void excp1(){
    int x=9;    int y=7;
    if(x-y==2)throw( 'A' );
    else cout<<"Its fine."<<endl;
}
int main(){
    try{        excp1();    }
    catch(char c){
        if(c=='B')cout<<"The diff is 2."<<endl;
        else throw;
    }
    return 0;
}

```

Output:

```

mouri@intell0-hub ~/c++-files/12-7 ./a.out
terminate called after throwing an instance of 'char'
[5] 12111 abort (core dumped) ./a.out

```

## 25. Write a class template to represent a generic vector. Include member functions to perform the following tasks:

- \* to create a vector
- \* to modify the value of a given element
- \* to multiply by a scalar.

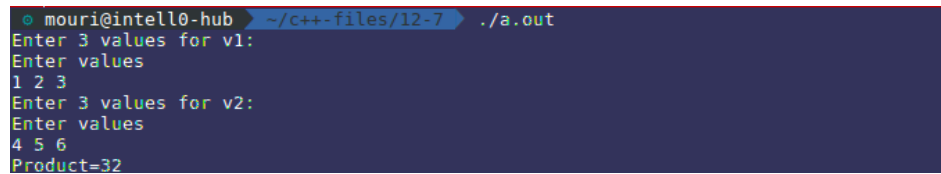
Program:

```

#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
template<class T>
class vector
{
    T *v;
public:
    void init(int size){
        v=new int [ size ];
        cout<<"Enter values"<<endl;
        for(int i=0;i<size;i++)cin>>v[ i ];
    }
    T multiply(vector &a,vector &b){
        T sum=0;
        for(int i=0;i<3;i++)sum+=a.v[ i ]*b.v[ i ];
        return sum;
    }
};
int main(){
    vector<int> v1;      vector<int> v2;
    vector<int> v;
    cout<<"Enter 3 values for v1:"<<endl;      v1.init(3);
    cout<<"Enter 3 values for v2:"<<endl;      v2.init(3);
    cout<<"Product="<<v.multiply(v1,v2)<<"\n";
    return 0;
}

```

Output:



```

mouri@intell0-hub ~/c++-files/12-7$ ./a.out
Enter 3 values for v1:
Enter values
1 2 3
Enter 3 values for v2:
Enter values
4 5 6
Product=32

```