B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Samarth Kumar Dubey
1BM22CS23

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

INDEX

Name Samarth Class 3E

1BM22CS235
Roll No. Subject College

Book name : Hamlet
Author name : William
Price : 400.
Number of pages : 230

Book name : IT
Author name : Stephen
Price : 300
Number of pages : 170

26/12/23

Samarth Kumar Dubey
1BM22CS235

```
        b[i] = new Books (name, author,
                      price, numPages);
    }
    for (i=0 ; i<n; i++)
    {
        System.out.println(b[i].toString());
    }
  }
}
```

Output

```
Enter number of books.
2
Enter name of book.
Hamlet
Enter price of book.
400
Enter number of pages in book.
230
Enter name of book
It
Enter price of book
300
Enter number of pages in book
170
Enter name of Author.
Stephen
```

Q Write Class Book, which Contains four members;
name, author, price, num Pages. Include
a constructor to set values for members.

26/12/23

Program

```
import java.util.Scanner;
class Books
{
    int price, numPages;
    String name, author;
    Books (String name, String author, int price,
          int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
public String toString( )
{
    String name, author, price, numPages;
     name = " Book name:" + this.name + "\n";
     author = " Author name:" + this.author + "\
```

Enter marks for subject 4:
92
Enter credits for subject 4:
2

Enter ~~credits~~ marks for subject 5:
80
Enter credits for subject 5:
1
Enter credits for subject 6:
80
Enter credits for subject 6:
1

Enter ~~credits~~ marks for subject 7:
80

Enter credits for subject 7:
1

Enter marks for subject 8:
80

Enter credits for subject 8:
1

Name Samarth
USN 1BM22CS235
SGPA 9.31578947

19-12-22

```
    {
          student   S1 = ~~neg~~ new  student();
           S1.getStudentDetails();
           S1.getMarks();
           S1.computeSGPA();
           System.out.Println("Name" +S1.name);
           System.out.Println ("USN" +S1.Usn);
           System.out.Println ("SGPA" +S1.sgpa);
    }
}
```

output

```
Enter your name
 Samarth
Enter your USN
1BM22CS235
enter marks for subject 1:
  80
Enter credits for subject 1:
  4
Enter ~~cat~~ marks for subject 2:
  80
Enter credits for subject 2:
  4
Enter ~~credits~~ marks for subject 3
  90
Enter credits for subject 3:
  3
```

/12/23

```
void  get marks ()
{
    for ( int  i=0 ; i<9 ; i++)
    {
        system.out. println ("Enter marks for
        subject" +(i+1) + ":");
        subject[i]. sub marks = s. nextInt();
        System.out. println ("Enter credits for
        subject " + (i+1) + ":");
        ~~**~~ subject [i]. credits = s. nextInt();
        subject [i]. grade =(subject [i].subMarks /10 +1);
        if  ( subject [i] .grade  >= 11)
                  subject[i]. grade= 10;
        if  (subject  [i] . grade < 4)
                  subject [i]. grade = 0;
    }
}
void  compute SGPA ()
{
    int  total credits = 0;
    for(int i=0 ; i<9 ; i++)
    {
        sgpa + = ( subject [i] . credits * subject[i]. grade);
        total credits + = subject [i]. credits;
    }
    sgpa = sgpa/ total credits;
}
}
class  main
{
    public static  void main (String args[])
```

17/12/23

Q. Develop a Java pgm. to create a class student with USN, name, an array credits, and an array marks. Include methods to accept & display and a method to calculate SGPA of a student.

$$\underline{SGPA} \Rightarrow \frac{\text{Course credits} \times \text{Creade points}}{\text{Course credits}}$$

```
import java.util.Scanner;
class subject
{
    int subMarks;
    int credits;
    int grade;
}
Class student
{
    subject subject[];
    String name, USN;
    double sgpa;

    Scanner s;
    Student()
    {
        int i;
        subject = new subject[9];
        s = new Scanner(System.in);
    }
    void get studentDetails()
    {
        System.out.Println("Enter your name");
        name = s.new();
        System.out.Println("Enter your USN");
        usn = S.next();
    }
```

```
            System.out.println(" \n Producer waiting
            wait();
        }
        catch (InterruptedException e){
            System.out.println(" \n Interrupted
            Consumer \n");
        }
        notify();
    }
}
class Producer Implements Runnable {
    Q q;
    Producer (Q q) {
        this.q = q;
        ~~~~new Thread (this, "Producer").sta
    }
    Public void run() {
        int i = 0;
        while (i<05) {
            q.put (i++);
        }
    }
}
Class consumer implements Runnable {
    Q q;
    Consumer (Q q) {
        this.q = q;
        new Thread (this, "Consumer").start();
    }
    public void run() {
        int i=0;
        while (i<05) {
            int r = q.get();
            System.out.println ("Consumed : "+r);
```

Lab 10

=) Demonstrate Inter process communication and deadlock

Pgm a : To demonstrate Inter process comm^n

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized    int get() {
                while (! valueSet)
                    try {
                            System.out.println ("\n Consumer waiting
                                                       \n");
                        wait();
                    }
                    catch (InterruptedException e)
                    {
                        System.out.println (" \n InterruptedException
                          caught ");
                    }
    System.out.println (" Got : "+n);
    valueSet = false;
    System.out.println (" \n Intimate Producer \n");
    notify();
    return n;
}
Synchronized    void put (int n) {
                while (value set)
                        try
                    {
```

Pgm for : To demonstrate deadlock

```
class A {
    synchronised void foo (B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + "Entered A.foo");
    try {
        ~~Threadsleep~~ Thread.sleep(1000);
    }
    catch (Exception e) {
        System.out.println("A interrupted");
    }
    System.out.println(name + "trying to call B.last()");
    b.last();
  }
  void last() {
      System.out.println("Inside A.last");
  }
}
class B {
    synchronised void bar (A a) {
        String name = Thread.currentThread().getName()
        System.out.println(Name + "entered B.bar"
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("Interrupted");
        }
```

```
Consumed : 1
Put : 2
Intimate Consumer
Producer waiting
Got : 2
Intimate producer
Consumed : 2
Put : 3
Intimate ~~Producer~~ Consumer
Producer waiting
Got : 3
Intimate Producer
Consumed : 3
Put : 4
Intimate Consumer
Got : 4
Intimate Producer
Consumed : 4
```

13-2-24

```
(++);
        }
    }
}
class PCFixed {
    public static void main (String args[]) {
            Q q = new Q();
          new Producer (q);
         new Consumer (q);
       System.out.println (" Press Control-c
       to stop.");
    }
}
```

Output

```
Press control-c to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
```

Output

Main.Thread entered A..foo

Racing.Thread entered B.bar

Racing Thread ~~entered~~ trying to call A.last().

Inside A.last

Back in other thread

main Thread trying to call B.last()

Inside A.last

Back in main thread,

11-2-24

```
            System.out.println (name + "Trying
            call A.last");
            a.last();
      }
      void last() {
         System.out.println ("Inside A.last");

      }
   }

class Deadlock implements Runnable
{
      A a = new A();
      B b = new B();
      Deadlock() {
            Thread.currentThread().setName("MainThre
            Thread t = new Thread (this, "RacingThre

            t.start();

            a.foo(b);

            System.out.println("Back in main thread


      }
   public void run() {
            b.bar(a);
            System.out.println("Back in other thread")


   }
public static void main(String args[]) {
            new Deadlock();
   }
}
```

```
Main Thread
Racing Threa
  Racing Th
  Inside A
  Back in
   Main T
   Inside
   Back
```

```
public class Printarea
{
        public static void main( String args[])
        {
              rectangle r = new rectangle ();
               r. printarea();
              triangle t = new triangle();
              t. print area ();
              Circle   c = new circle ();
              c. printarea()
        }
}
```

output :

```
Enter  length  and  breadth of rectangle
 10
 20
~~[illegible]~~ length  of rectangle : ~~[illegible]~~ 10   breadth of rectangle : 20
 area  of rectangle  : 200.0
Enter  base  ands  height  of ~~[illegible]~~ triangle
 12

 2
 height  of  triangle : 12        base of triangle : 2
area  of  triangle  : 12.0
Enter  radius  of  circle
2
 radius  of   circle : 2
 area  of circle : 12.56
```

2-1-24

```
            rectangle );
        a = s.nextInt();
        b = s.nextInt();
       System.out.println("lengt of the
    rectangle : "+a+ "breadth of rectangle: "+b);
    System.out.println("Area of rectangle "+area_rect);
  }
}

class Triangle extends shape
{
    public double area_triangle;
    public void printarea()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter base and
        height of triangle");
        a = s.nextInt();
        b = s.nextInt();
        area_triangle = (a*b)/2 ;
        System.out.println("height of triangle : "+a+
        "base of triangle "+b);
        System.out.println("area of triangle "+
        area_triangle);
    }
}
```

1/1/24

Develop a Java program to create an abstract class Shape containing two integers and an empty method printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class shape. Each one of the classes contain only the method printArea() that prints area of given shape.

Step

1) Add class InputScanner
2) Add class shape extend InputScanner
3) Add class shape extends ~~Input~~ shape

```
import java.util.Scanner;
abstract class Shape
{
    int a,b;
    abstract public void printarea();
}
class rectangle extends shape
{
    public double area_triangle;
    public void printarea()
    {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter length & breadth of
```

vii) Substring is matched

viii) true
true

ix) false
true

x) true
false

xi) apple ball cat dog ent free gem
hen ice jug kite man net orange parrot
queen ring sun tree

xii) 1 2 3 4 5 6 7 8 9 10

xiii) This is a Test , This is too

xiv) Hello world

xv) Carnage

xvi) Hello friends

xvii) Student :

name : Anu
Regno : 123
sem : 3
CGPA : 9.6

```
ca. withdraw (amount 2);
ca. checkmin ();
break;

case 3: ca. display;
        break;

case 4 : System. exit (0);

    }
  }
}
```

Output

```
Enter name : Joe
Enter type : ~~saving~~ current
Enter account numb : 1001
Enter initial balance : 1000

Menu :
1 Deposit  2 withdraw  3 display
1
Enter amount : 1000

Menu :
1 Deposit  2 withdraw  3 display
2
Enter amount : 1600
Balance is less than min balance, service charge 50
Balance is : 950
```

16-1-24

Java String programs

i) Demonstrate various string constructors

output

String constructed by character array abcdef

String by character array with index bcd

String constructed with String object abcdef

String with ASCII characters ABCD

String

ii) Demonstrate String length, String concat

output

Concatenated string s1 & s2 is abcdef ABCD

and length of both strings are 6 & 3

iii) a=333

iv) BMSCE

v) 65

66

67

abc

true

false

false

true

```
        System.out.println("Underflow");
         return null;
        }
      else
        return stk[top--];
      }

   }
Public Class TestStack {
          public static void main(String args[])
          {
            Stack <Integer> S1 = new Stack <Integer>
                                           ();
            Stack <Double> S2 = new Stack <Double>();
            Scanner S = new Scanner(System.in);
            System.out.println("Enter elements in
            integer stack");
            for(int i=0; i<5; i++){
                int n=S.nextInt();
                 S1.push(n);
              }
            System.out.println("Enter element
             in Double stack");
             for (i=0; i<5; i++){
                  double m= S.new Double();
                  S2.push(m);
              }
```

```
System.out.println ("Elements of S2)
for (int i=0 ; i<5 ; i++){
        System.out.println (S2.pop());
    }
    S.close()
  }
}
```

<u>Output</u>

```
Enter elements in integer stack
 1  3  6  9  12
Enter elements in double stack
 2  4  5  7  8
 Elements in S1
  12
  9
  6
  3
  1
Elements in S2
  8.0
  7.0
  5.0
  4.0
  2.0
```

ii) For ~~savings account~~

~~Class~~ SavingAccount ~~extends~~ Account {

```
import java.util.Scanner;
Class account
{
   String name;
   int accno;
   String type;
  double balance;

 account ( String name, int accno, String type, double balan)
 {
          this.name = name;
          this.accno = accno;
          this.type = type;
          this.balance = balance;
 }
 void deposit ( double amount)
 {
        balance += amount;
 }
{ void withdraw (double amount)
     if ((balance - amount ) >= 0)
```

Student 2

Name : Samart
Reg no : 141
Sem : 3
CGPA : 9.6

xviii) Char at 3 is 'x'
abc
reverse : incha

xix) Eagle is flying
Eagle makes a sound

Hawk is flying
Hawk makes a sound

xx) Carea : 28.26

cper : 18.84

Tarea : 90

Tper : 82

Name: Samarth Kumar Duby

USN : 1BM22CS235

Generic

Write a java program to create a generic class stack which holds 5 integers & 5 doubles

```
import java.util.*;
    class stack <T> {
          T stk[ ];
          int top;
          int size=10;
       stack( ) {
                   stk = (T[ ]) new
                          Object [size];
                   top = -1;
              }

    void push( T item) {

             if (top == size-1) {
                  System.out.println("overflow");
                  }
             else {
                      stk[++top] = item;
                  } }
 T pop(){
       if (top < 0) {
```

```
System.out.println("Enter account number: ");
int accno = s.nextInt();

double amount1, amount2;
account ac = new ~~Sav~~ Account(name, accno,
                type, balance);

SavAcct sa = new SavAcct(name, accno, balance
CurAcct ca = new CurAcct(name, accno, balance

while (true)
{
    if (acc.type.equals("Savings"))
    {
        System.out.println("\n Menu \n 1. Deposit
        \n 2 Withdraw \n 3. compute Interest
        4. display");
        System.out.println("Enter choice: ");
        ch = s.nextInt();
        switch (ch)
        {
            Case 1: System.out.print("Enter
                    amount: ");
                    amount1 = s.nextInt();
                    sa.deposit(amount1);
                    break;
```

```
Case 2
Case
Case
Case
defa
}
}
else
{
Syst
2
Syst
Ch
Swi
{
```

```
Case 2 : System. out. println ("Enter amount");
            amount 2 = S. next Int();
            Sa. withdraw (amount 2);
            break;

    Case 3 :    Sa. interest ();
                 break;

    Case 4 :    Sa. display ();
                 break;

    Case 5 :  System. exit (1);
    default : System. out. println ("Invalid input");
                  break;
    }

3
else
{
    System. out. println ("\n Menu \n 1. Deposit
     2. withdraw \n 3. Display");~~\n~~ .
    System . out . println (" Enter the choice : ");
    ch = S. next Int ();
    Switch (ch)
    {
        case 1 : System. out. println ("Enter the
        amount : ");
          amount 1 = S. next Int();
           ca. deposit (amount);
            brea
          case 2 :   System. out. println ("Enter the
                 amount : ");
                 amount 2 = S. next Int () ;
        balance = amount;
    else
        System.out.print(" Insufficient balance");
                                              }
    }
 void display
 {
    System.out.println(" name :" +name+
                 " acc no: " +acc no+
              "type : " + type );
 }
}
Class savAcct extends Account
{
    private static double rate =5;
    savAcct (String name, int accno, double
                 balance)
    {
        super (name, accno, "savings", balance);
    }
   void interest()
   {
      balance += balance *(rate)/100 ;
      System.out.println(" balance: " + balance);
   }
Class curAcct extends account
{
    private double minBal =500;
    private double serviceCharge = 50;
```

```
        ( String name, int accno, double
              balance)
  {
        super (name , accno, "current" , balance);
  }

  void    Check min ( )
  {
        if  (balance < min Bal)
        {
            System.out.println (" Balance is less
         than min balance , service charge
         imposed : " + service Charges);
         balance - = service Charge ;
         System.out.println (" Balance is    " + balance);
        }
  }
}

Class   account main
{
   public static void main ( String args [ ] )
   {
        Scanner  s = new Scanner ( System.in );
        System.out.println (" Enter the name: ");
        String name = s. next( );
         System.out.println (" Enter type" );
         String  type = s.next ( );
```

```
                System.out.println(" Enter de
                of student");
                name = S.next();
                usn = S.next();
                sem = S.nextInt();

            }

      public void displayStudentdetails() {
            System.out.println(" Name " + name
                               + "\n USN " + usn
                               + " \n sem: " + sem);
      }

  }
```

Internals.java

```
Package CIE;
import java.util.Scanner;
public class Internals extends Student {
        protected int marks[] = new int[5];
        public void inputCIEmarks()
      {
            Scanner s = new Scanner(System.in);
            System.out.println(" Enter marks of
            students");
            for (int i=0; i<5; i++)
            {
                marks[i] = s.nextInt();
            }
        }
    }
```

External.java

```
  package SEE;

  import CIE.Internals;
  import java.util.Scanner;
  public External class Externals extends Internals {
          protected int marks[];
          protected int finalmarks[];
      public Externals() {
              marks = new int[5];
              finalmarks = new int[5];
      }

      public void inputSEEmarks() {
              Scanner s = new Scanner(system.in);
              for( int i=0 ; i<5 ; i++)
              {
                  System.out.println("Subject "+(i+1) +
                                          " Marks:");
                  marks[i] = s.nextInt();
              }
      }
```

o Create a package called maths having a class number (add & subtract method). Implement a simple class called Maths Demo to use maths (outside package maths) that makes use of package provided by maths.

Q Create a package CIE which has two classes Student and Internals. The class student has members like usn, name, sem. The class internals derived from student has an array that store SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses

Student. java

```
Package CIE;
impor java. util. Scanner;
public class Student { protected String usn = ~~new~~
    new String();
    protected String name = new String();
    protected int sem;
public void inputStudentDetails () {
                                   Scanner s = new
                                   Scanner( System. in);
```

```
Subject 4 marks : 4
Subject 5 marks : 44

Enter std marks
Subject 1 marks : 68
Subject 2 marks : 75
Subject 3 marks : 87
Subject 4 marks : 94
Subject 5 marks : 83
```

Displaying data :

```
Subject 1 marks : 81
Subject 2 marks : 82
Subject 3 marks : 91
Subject 4 marks : 92
Subject 5 marks : 82
```

24-1-24

```
public void calculate Final Marks () {
        for ( int i=0 ; i<5 ; i++ ).
        {
            final Marks [i] = marks(i)/2
                            + Super. marks[i]/2;
        }
}
public    void display Final Marks () {
        display Student Details ();
        for (int i=0 ; i<5 ; i++)
            System.out.println ("  Subject " +
                        (i+1) + " : " +
                    final Marks [i] );
}
```

Main. java

```
import SEE . Externals ;

class main {

    public static void main ( String args[] ) {

      int   num of Students  = 2 ;

      External  final Marks [] = new Externals [num of Stud]

   for ( int i=0 ; i< num of Studen ; i++)
        {
          final Marks[i] = new External ();
```

System.

}

}

Output

Enter

Enter

Enter

Enter

Enter

Subje

subj

subj

```
                    final Marks [i]. input Student Details ();
                        System. out. println ("Enter CIE Marks");
                        Finalmarks [i]. input CIE Marks ();
                        System. out. println ( "Enter SEE marks");
                        final Marks [i]. Input SEE marks ();
                  }

      System. out. println ("Displaying data: \n");
              for (int i=0 ; i< num of Student ; i++)
              {
                    final Marks [i]. Calculate Final Marks();
                     final Marks [i]. display Final Marks ();
              }
      }
}
```

Output

```
Enter USN : 1BM22CS235
Enter Name : Samarth
Enter semester : 2
Enter CIE marks
Enter Internal marks
 Subject 1 marks : 47
 subject 2 marks : 44
 subject 3 marks : 48
```

```
java ExceptionAge
Enter Age of Father
 20
Enter age of Son
 20
Son cannot be same age as Father
 Wrong Age

java. ExceptionAge

Enter age of Father
 30
Enter age of Son
 5

son's age = 5
Father's ~~ag~~ age = 30
```

30-1-14

Lab 7

Write a program that demonstrates handling of exception in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implements a constructor which takes the age and throws the exception WrongAge() when input age<0

```
import java.util.Scanner;

class WrongAge extends Exception{
        public WrongAge ~~extends Exception~~ String (s) {
            System.out.println(s);
    }
}

class Father {
  int age;
  public Father (int age) {
     this.age = age;
  }
```

```
            catch ( Wrong Age
              {
                System . out. println (e);
              }
         }
    }
}
```

Output

```
Enter age of Father
50
Enter age of Son
-10
Age cannot be negative
Wrong Age

java ExceptionAge

Enter age of Father
20
Enter age of Son
30
Son cannot be older than Father
Wrong Age
```

```
            if (age < 0)
            {
                throw new WrongAge ("
                Age cannot be negative.");
            }
    }

    public void display () {
        System.out.println ("Father's age = " + age);
    }
}

class Son extends Father {
    public Son (int age) {
        super (age);
    }
    int sage = age;
    public void Checkage2 () throws WrongAge {
        if (sage < 0)
        {
            throws new WrongAge ("Age cannot
                be negative.");
        }
        else if (sage > 0)
        {
            throw new WrongAge (" ~~Age cannot~~
            Son cannot be older than father
```

```
else if ( sage == age)
{     throw new WrongAge (" Son connot be
      same age as father ");

}

}
public void display () {
      System.out.println (" Son's age : " +sage);

}

}
Public class AgeException {
      public static void main ( String [] args) {
           Scanner sc = new Scanner (System.in);
           System.out.println (" Enter age of father");
           Father f = new Father ( sc.nextInt());
           System.out.println (" Enter age of Son");
           Son s = new Son ( Sc.nextInt());
           sc.close();
           S.age = f.age;
           try {
                  f.checkage1();
                  S.checkage2();
                  S.display();
                  f.display();
           }
```

QuadraticMain.java

```
import java.util.*;
class Quadratic
{
int a, b, c;
double r1, r2, d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b,c");
a = s.nextInt();
b = s.nextInt();
c = s.nextInt();
}
void compute()
{
while(a==0)
{
System.out.println("Not a quadratic equation");
System.out.println("Enter a non zero value for a:");
Scanner s = new Scanner(System.in);
a = s.nextInt();
}
d = b*b-4*a*c;
if(d==0)
{

r1 = (-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Roo1 = Root2 = " + r1);
}
else if(d>0)
{
r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
}
else if(d<0)
{
System.out.println("Roots are imaginary");
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = " + r1 + " + i"+r2);
System.out.println("Root1 = " + r1 + " - i"+r2);
}
}
}

class Quadraticmain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
System.out.println("NAME:Samarth");
System.out.println("USN:1BM22CS235");
}
}
```

Main.java

```
//NAME:Samarth

import java.util.*;
class Subject{
	int submarks;
	int credits;
	int grade;
}

class Student{
	Subject subject[];
	String name;
	String USN;
	double SGPA;
	Scanner s;

	Student()
	{
	int i;
	subject=new Subject[9];
	for(i=0;i<9;i++)
		subject[i]=new Subject();
	s=new Scanner(System.in);
	}

	void getstudentdetails()
	{
	System.out.println("enter name and usn");
	this.name=s.nextLine();
	this.USN=s.nextLine();
	}

	void getmarks()
	{
	int i;
	for(i=0;i<8;i++)
	{
		System.out.println("Enter the marks of " + (i+1)+ " Subject");
		subject[i].submarks=s.nextInt();
		System.out.println("Enter the credits of " + (i+1)+ " Subject");
		subject[i].credits=s.nextInt();
		subject[i].grade=(subject[i].submarks/10)+1;
		if(subject[i].grade>10){
			subject[i].grade=10;
```

```
        }
        if(subject[i].grade<4){
            subject[i].grade=0;
        }
    }
    }

    void computeSGPA()
    {
        int totalcredits=0;
        int sum=0;
        int i;
        for( i=0;i<8;i++)
        {
            sum=sum+subject[i].grade * subject[i].credits;
            totalcredits=totalcredits+subject[i].credits;
        }
        this.SGPA=(double) sum/totalcredits;

    }
}

public class Main
{
    public static void main(String args[])
    {
        Student s1=new Student();
        s1.getstudentdetails();
        s1.getmarks();
        s1.computeSGPA();
        System.out.println("NAME:"+s1.name);
        System.out.println("USN:"+s1.USN);
        System.out.println("SGPA:"+s1.SGPA);
    }
}
```

BooksMain.java

```
import java.util.*;

class Book{
String name;
String author;
int price;
int numpages;

Book(String name,String author,int price,int numpages)
{
    this.name=name;
    this.author=author;
    this.price=price;
    this.numpages=numpages;
}
```

```
 String getname()
{
        return name;
}
 String getauthor()
{
        return author;
}
 int getprice()
{
        return price;
}
int getnumpages()
{
        return numpages;
}

void setname(String name)
{
        this.name=name;
}
void setauthor(String author)
{
        this.author=author;
}
void setprice(int price)
{
        this.price=price;
}
void setnumpages(int numpages)
{
         this.numpages=numpages;
}

public String toString()
{
        String name,author;
        String price,numpages;
        name="Book name:" + this.name + "\n";
        author="Author name:"+ this.author + "\n";
        price="Price:"+ this.price + "\n";
        numpages="Number of pages:" + this.numpages+ "\n";
        return name+author+price+numpages;
}
};

public class BooksMain{
        public static void main(String args[])
        {
                Scanner s=new Scanner(System.in);
                int n,price,numpages;
                String name, author;
                System.out.println("Enter n");
                n=s.nextInt();
                Book b[];
                b=new Book[n];
                System.out.println("By toString method");
                for(int i=0;i<n;i++)
                {
```

```
            System.out.println("Enter the book name");
            name=s.next();
            System.out.println("Enter the author name");
            author=s.next();
            System.out.println("Enter the price of book");
            price=s.nextInt();
            System.out.println("Enter the number of pages");
            numpages=s.nextInt();
            b[i]=new Book(name,author,price,numpages);
        }
        for(int i=0;i<n;i++)
        {
            String Bookdetails=b[i].toString();
            System.out.println(Bookdetails);
        }
        System.out.println("**********************************");
        System.out.println("By get and set methods");
        for(int i=0;i<n;i++)
        {

            System.out.println("enter book name:");
            name=s.next();
            b[i].setname(name);

            System.out.println("enter author name:");
            author=s.next();
            b[i].setauthor(author);

            System.out.println("enter book price:");
            price=s.nextInt();
            b[i].setprice(price);

            System.out.println("enter number of pages:");
            numpages=s.nextInt();
            b[i].setnumpages(numpages);
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("Book name:"+b[i].getname());
            System.out.println("Author name:"+b[i].getauthor());
            System.out.println("Book price:"+b[i].getprice());
            System.out.println("Number of pages:"+b[i].getnumpages());
        }
        System.out.println("NAME:Samarth");
        System.out.println("USN:1BM22CS235");
    }
}
```

AbstractMain.java

```
import java.util.Scanner;

class InputScanner{
	Scanner s;
	InputScanner(){
		s = new Scanner(System.in);
	}
}
abstract class Shape extends InputScanner{
	double a;
	double b;

	abstract void getInput();
	abstract void displayArea();
}

class Rectangle extends Shape{

	void getInput(){
		InputScanner is = new InputScanner();
		System.out.println("Enter the length and breadth of the rectangle :");
		a = is.s.nextDouble();
		b = is.s.nextDouble();
	}
	void displayArea(){
		System.out.println("The area of the rectangle is :"+(a*b));
	}


}
class Triangle extends Shape{

	void getInput(){
		InputScanner is = new InputScanner();
		System.out.println("Enter the base and height of the triangle :");
		a = is.s.nextDouble();
		b = is.s.nextDouble();
	}
	void displayArea(){
		System.out.println("The area of the triangle is :"+(a*b*0.5));
	}


}
class Circle extends Shape{

	void getInput(){
		InputScanner is = new InputScanner();
		System.out.println("Enter radius of the Cirlce :");
		a = is.s.nextDouble();
	}
	void displayArea(){
		System.out.println("The area of the Circle is :"+(3.14*a*a));
	}


}
public class AbstractMain{
	public static void main(String args[]){
		System.out.println("HI");
		Rectangle rect = new Rectangle();
		rect.getInput();
```

```
                rect.displayArea();

                Triangle triangle = new Triangle();
                triangle.getInput();
                triangle.displayArea();

                Circle circle = new Circle();
                circle.getInput();
                circle.displayArea();
                System.out.println("NAME : Samarth");
                System.out.println("USN : 11BM22CS235");
        }
}
```

Additional.java

```
import java.util.*;
class Students{
        String USN;
        String Name;
        int s1,s2,s3;
        double avg;

        void getDetails(){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the USN:");
                USN = sc.next();
                System.out.println("Enter the Name of student:");
                Name= sc.next();
                System.out.println("Enter all three subject marks");
                s1 =sc.nextInt();
                s2 =sc.nextInt();
                s3 =sc.nextInt();
        }
        void marks(){
                if(s1 >= s2 && s3 >= s2){
                        avg = (double)((s1+s3)/2);
                }
                else if(s1 >= s3 && s2 >= s3){
                        avg = (double)((s1+s2)/2);
                }
                else{
                        avg = (double)((s3+s2)/2);
                }
        }
        void display(){
                marks();
                System.out.println("USN : "+USN);
                System.out.println("NAME : "+Name);
                System.out.println("Avg marks : "+avg);
        }
        protected void finalize(){
            System.out.println("finalize() method called");
        }
}


public class Additional{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of students:");
        int n = sc.nextInt();
        Students s[]  = new Students[n];

        for(int i = 0;i<n;i++){
            s[i] = new Student();
            s[i].getDetails();
            s[i].display();
        }
        System.out.println("NAME : SHIVARAJ K PUJARI");
        System.out.println("USN : 11BM22CS259");
    }
}
```

Bank.java

```
import java.util.*;
class Account{
    String name;
    int accno;
    String acctype;
    double balance;
    Account(String name,int accno,String acctype,double balance)
    {
        this.name=name;
        this.accno=accno;
        this.acctype=acctype;
        this.balance=balance;
    }
    void deposit(double amt){
        balance+=amt;
    }
    void withdraw(double amt){
        if((balance-amt)>=0){
            balance-=amt;
        }
        else{
            System.out.println("Insufficient balance");
        }
    }
    void display(){
        System.out.println(" Name:" + name + " accnor:" + accno + " account_type:" + acctype + "
balance:" + balance);
    }
}
class Savingaccount extends Account{
    private static double rate=5;
    private double minbal=500;
    Savingaccount(String name,int accno,double balance){
        super(name,accno,"savings",balance);
    }
    void interest(){
        balance+=balance*(rate)/100;
        System.out.println("Balance:" + balance);
    }
    void checkmin(){
        if(balance<minbal){
            System.out.println("balance is less than minimum balance,insufficient balance");
        }
    }

}
class Curracc extends Account{
    private double minbal=500;
    private double charge=50;
    Curracc(String name,int accno,double balance){
        super(name,accno,"current",balance);
    }
    void checkmin(){
        if(balance<minbal){
            System.out.println("balance is less than minimum balance,service charges imposed:" + charge);
            balance-=charge;
            System.out.println("Balance is:" + balance);
        }
    }
}

class Bank{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the name:");
        String name=sc.nextLine();
        System.out.println("Enter the type of account:");
        String type=sc.nextLine();
        System.out.println("Enter the account number:");
        int accno=sc.nextInt();
        System.out.println("Enter the Balance:");
        double balance=sc.nextDouble();
        int ch;
        double amt1,amt2;
        Account ac=new Account(name,accno,type,balance);
        Savingaccount sa=new Savingaccount(name,accno,balance);
        Curracc ca=new Curracc(name,accno,balance);
        System.out.println("NAME:Shivaraj K Pujari");
        System.out.println("USN:1BM22CS259");
        while(true)
        {
            if(ac.acctype.equals("savings"))
            {
                System.out.println("\nMENU\n 1.deposit 2.withdraw 3.compute interest 4.diplay 5.exit");
                System.out.println("Enter the choice");
                ch=sc.nextInt();
                switch(ch)
                {
                    case 1:System.out.println("Enter the amount");
                        amt1=sc.nextInt();
                        sa.deposit(amt1);
                        break;
                    case 2:System.out.println("Enter the amount");
                        amt2=sc.nextInt();
                        sa.withdraw(amt2);
                        sa.checkmin();
                        break;
                    case 3:sa.interest();
                        break;
                    case 4:sa.display();
                        break;
                    case 5:System.exit(0);

                    default:System.out.println("Inavlid input");
                        break;
                }
            }
            else{
                System.out.println("\n MENU \n 1.deposit 2.withdraw 3.display ");
                System.out.println("Enter the choice");
                ch=sc.nextInt();
                switch(ch){
                    case 1:System.out.println("Enter the amount:");
                    amt1=sc.nextInt();
                    ca.deposit(amt1);
                    break;
                    case 2:System.out.println("Amount to be withdraw");
                    amt2=sc.nextInt();
                    ca.withdraw(amt2);
                    ca.checkmin();
                    break;
                    case 3:ca.display();
                    break;
                    case 4:System.exit(0);
                }
            }
        }

    }
}
```

Student.java

```
package CIE;
import java.util.*;
public class Student {
    protected String name=new String();
    protected String usn=new String();
    protected int sem;
    public void inputstudentdetails()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the student's Name :");
        name=s.nextLine();
        System.out.println("Enter USN:");
        usn=s.nextLine();
        System.out.println("Enter semester:");
        sem=s.nextInt();
    }
    public void display()
    {
        System.out.println("Student details:");
        System.out.println("NAME:" + name);
        System.out.println("USN:"+ usn);
        System.out.println("SEM:"+ sem);
    }
}
```

Internals.java

```
package CIE;
import java.util.*;
public class Internals extends Student{
    protected int marks[]=new int[5];
    public void inputcie()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the Marks of 5 subjects in CIE:");
        for(int i=0;i<5;i++)
        {
            System.out.print("Subject " + (i+1) + " marks:");
            marks[i]=s.nextInt();
        }
    }
}
```

Externals.java

```
package SEE;
import CIE.Internals;
import java.util.*;
public class Externals extends Internals{
    protected int marks[];
    protected int finalmarks[];
    public Externals()
    {
        marks=new int[5];
        finalmarks=new int[5];
    }
    public void inputsee()
    {
        Scanner s=new Scanner(System.in);
        for(int i=0;i<5;i++)
        {
            System.out.print("Subject " + (i+1) + " marks:");
            marks[i]=s.nextInt();
        }
    }
    public void cfmarks()
    {
        for(int i=0;i<5;i++)
        {
            finalmarks[i]=marks[i]/2 + super.marks[i];
        }
    }
    public void displayfmarks()
    {
        display();
        for(int i=0;i<5;i++)
        {
            System.out.println("subject"+ (i+1) +":"+ finalmarks[i]);
        }
    }
}
```

StudentMain.java

```
import SEE.Externals;
public class StudentMain {
    public static void main(String[] args) {
        int n=1;
        Externals finalmarks[]=new Externals[n];
        for (int i = 0; i < n; i++)
        {
            finalmarks[i]=new Externals();
            finalmarks[i].inputstudentdetails();
            System.out.println("Enter CIE marks:");
            finalmarks[i].inputcie();
            System.out.println("Enter SEE marks");
            finalmarks[i].inputsee();
        }
        System.out.println("Displaying data:");
        for(int i=0;i<n;i++)
        {
            finalmarks[i].cfmarks();
            finalmarks[i].displayfmarks();
        }
    }
}}
```

TestStack.java

```
import java.util.Scanner;

class Stack<E> {
    E stck[];
    int top;
    final int SIZE = 10;

    Stack() {
        stck = (E[])new Object[SIZE];
        top = -1;
    }

    void push(E item) {
        if (top == SIZE - 1)
            System.out.println("Stack is full");
        else
            stck[++top] = item;
    }
    E pop() {
        if (top < 0) {
            System.out.println("Stack underflow");
            return null;
        } else
            return (E) stck[top--];
    }
}

public class TestStack {
    public static void main(String[] args) {
        Stack<Integer> mystack1 = new Stack<Integer>();
        Stack<Double> mystack2 = new Stack<Double>();

        Scanner s = new Scanner(System.in);

        System.out.println("Enter elements into the Integer stack");
        for (int i = 0; i < 5; i++) {
            int n = s.nextInt();
            mystack1.push(n);
        }

        System.out.println("Enter elements into the Double stack");
        for (int i = 0; i < 5; i++) {
            double m = s.nextDouble();
            mystack2.push(m);
        }

        System.out.println("Elements of stack 1 ");
        for (int i = 0; i < 5; i++)
            System.out.println(mystack1.pop());

        System.out.println("Elements of stack 2 ");
        for (int i = 0; i < 5; i++)
            System.out.println(mystack2.pop());

        s.close();
    }
}
```

SpMain.java

```
import java.util.*;
class Wrongage extends Exception{
		public Wrongage(String s)
		{
				super(s);
		}
}
class Father{
		int fage;
		Father () throws Wrongage
		{
				System.out.println("enter father's age");
				Scanner s=new Scanner(System.in);
				fage=s.nextInt();
				if(fage<0)
```

```
        {
            throw new Wrongage("Age cannnot be negative");
        }
    }
    void display()
    {
        System.out.println("Father's Age is :" + fage);
    }
}
class Son extends Father{
    int sage;
    Son () throws Wrongage
    {
        System.out.println("enter son's age");
        Scanner s=new Scanner(System.in);
        sage=s.nextInt();
        if(sage>fage)
        {
            throw new Wrongage("Son's age cannot be greater than father's age");
        }
        else if(sage==fage){
            throw new Wrongage("Age cannnot be same");
        }
        else if(sage<0)
        {
            throw new Wrongage("Age cannnot be negative");
        }
    }
    void sdisplay()
    {
        System.out.println("Son's Age is :" + sage);
    }
}
class Spmain{
    public static void main(String args[])
    {
        try{
            Son s=new Son();
            s.display();
            s.sdisplay();
        }
        catch(Wrongage e){
            System.out.println(e);
        }
        System.out.println("NAME: Samarth");
        System.out.println("USN:1BM22CS235");
    }
}
```

Tmain.java

```
class CS extends Thread
{
public void run()
{
for(int i=1; i<=20; i++){
    try{
```

```
                System.out.println("BMS College of Engineering" + i);
                Thread.sleep(10000);
        }
        catch(InterruptedException e){
                System.out.println("thread error");
        }
}
}
}

class IS extends Thread
{
public void run()
{
for(int i=1; i<=20; i++){
        try{

                System.out.println("Computer Science " + i);
                Thread.sleep(2000);
        }
        catch(InterruptedException e){
                System.out.println("thread error");
        }
}
}
}

public class Tmain {
public static void main(String args[])
{
CS c1 = new CS();
c1.start();

IS i1 = new IS();
i1.start();
System.out.println("NAME:Samarth");
System.out.println("USN :1BM22CS235");
}
}
```

SwingDemo.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
  SwingDemo(){
    // create jframe container
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    // to terminate on close
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // text label
    JLabel jlab = new JLabel("Enter the divider and divident:");
    // add text field for both numbers
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);

    // calc button
    JButton button = new JButton("Calculate");

    // labels
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();

    JLabel anslab = new JLabel();

    // add in order :)
    jfrm.add(err);  // to display error bois
    jfrm.add(jlab);
    jfrm.add(ajtf);
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(blab);
    jfrm.add(anslab);

    ActionListener l = new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            System.out.println("Action event from a text field");
        }
    };
    ajtf.addActionListener(l);
    bjtf.addActionListener(l);

    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            try{
                int a = Integer.parseInt(ajtf.getText());
                int b = Integer.parseInt(bjtf.getText());
                int ans = a/b;

                alab.setText("\nA = " + a);
                blab.setText("\nB = " + b);
                anslab.setText("\nAns = "+ ans);
            }
            catch(NumberFormatException e){
                alab.setText("");
                blab.setText("");
              anslab.setText("");

                err.setText("Enter Only Integers!");
            }
            catch(ArithmeticException e){
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });
```

```
        // display frame
        jfrm.setVisible(true);
    }

    public static void main(String args[]){
        // create frame on event dispatching thread
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
                new SwingDemo();
            }


        })
System.out.println("NAME:Samarth");
                System.out.println("USN:1BM22CS235");
    }
}
```