1. Quadratic equation using Java.

```
import. java. util. Scanner;
public class quadratic
{ public static void main ( String [] args)
 { scanner input = new Scanner (system.in);
  System.out.print ("Enter value of a :");
   double a = input . nextDouble ();
  System. out. print ("Enter value of b :");
   double b = input . nextDouble();
   System. out . print ("Enter value of c :");
    double c = input . next Double();
double d = (b*b) -(4*a*c);
 if (d > 0)
 { double r1 = (-b + Math.pow(d,0.5))/(2*a);
   double r2 = (-b + Math.pow(d,0.5))/(2*a);
System.out.println ("Roots are real and
                    distinct and are " +r1+
                  " and " + r2);
}
else if (d==0)
 { double r1 = -b/(2*a);
System.out.println ("Roots are real and
                  equal and is " +r1);
}
else
{ System.out.println ("Roots are imaginary");
}
}
}
```

2. Create a class Book containing author, price, name. Include methods to set & get the details. Develop a java program for n books.

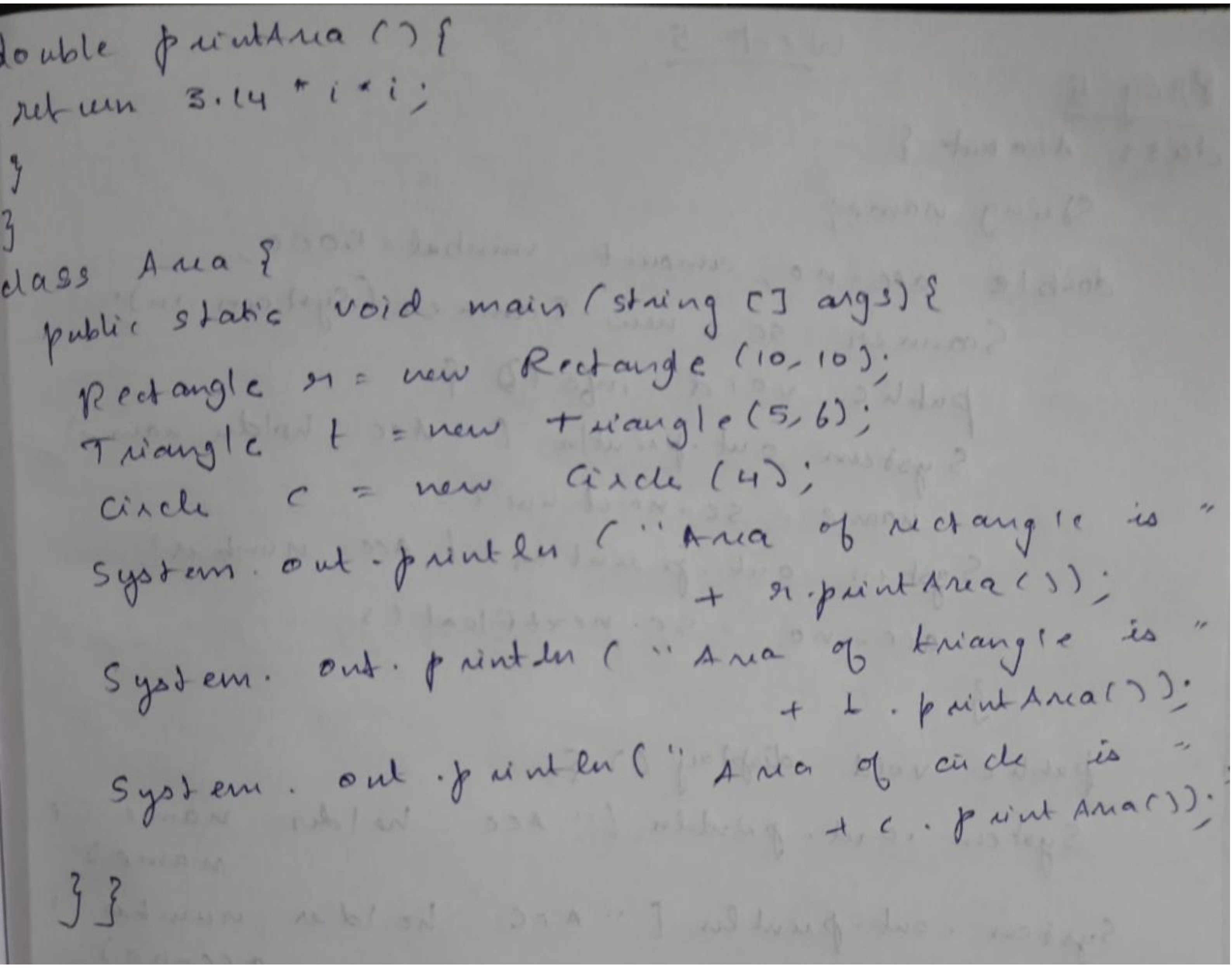
```
import java.util.*;
import java.io.*;

class Book {
 String title, author;
  double price;
  int numPages;
Book () {
    title = "Default";
    author = "Default";
    price = 0.0;
   numPages = 0;
}
void setTitle (String t) {
   title = t;
 }
void setAuthor (String a) {
    author = a;
 }
void setPrice (double p) {
   price = p;
 }
void setPages (int np) {
   numpages = np;
 }
public string toString () {
  return title + author + price + numPages;
 }
```

```
class BookDetails {
  public static void main (String args[]) {
    String t, a;
    double p;
    int np, n;
  Scanner sc = new Scanner (System.in);
  System.out.println ("Enter books number
   n = sc.nextInt();
  for (int i=0; i<n; i++)
  {  System.out.println ("Enter Title Books");
       t = sc.next();
     System.out.println ("Enter Author of
     Books");
     a = sc.next();
    b[i] = new Book();
    b[i].setTitle(t);
    b[i].setAuthor(a);
    b[i].setPrice(p);
    b[i].setPages(np);
  }
   System.out.println ("Title \t Author \t Pri
                      Pages\n");
   for (int i=0; i<n; i++) {
       System.out.println(b[i]);
   }
  }
}
```

Output

Q. Develop a Java Program to create an abstract class named shape that contains 2 integers and printArea(). Provide 3 classes named Rectangle, Triangle & Circle as subclass.

```
abstract class shape {
 double i, j;
 shape (double x, double y) {
  i = x;
  j = y;
 }
 shape (double r) {
  i = r;
 }
abstract
~~double~~ printArea () {
 return 0;
 }
}
class Rectangle extends Shape{
 Rectangle (double l, double b) {
  super( l, b);
  }
  double printArea() {
  return i * j;
 }
}
class Triangle extends shape {
 Triangle (double l, double h) {
   super (l, h);
 }
 double printArea () {
  return i*j/2;
 }
}
class circle extends shape {
 Circle (double r) {
  super(r); }
```



```
double printArea (){
return 3.14 * i * i;
}
}
class Area {
 public static void main (string [] args){
  Rectangle r = new Rectangle (10, 10);
  Triangle  t  = new triangle (5, 6);
  Circle   c  = new  Circle (4);
  System.out.println ("Area of rectangle is "
                         + r.printArea());
  System. out. println ( "Area of triangle is "
                         + t . printArea());
  System . out .println (" Area of circle is "
                         + c . print Area());
 }}
```

Week 5

Prog 5

```
class Account {
    String name;
  double acc-no, amount . minbal = 5000;
      Scanner sc = new Scanner (System.in);
       public void info () {
       System out.println (" Acc holder name);
         name = sc.nextLine ();
      System . out . println ( acc. number);
        acc-no  = sc. nextFloat ();
      }
  public void display () {
   System .out. println (" Acc . holder  name " +
                                        name );
 System . out-println [ " Acc  holder  number "+
                                     acc-no );
 system . out-println ( "Balance   " + amount );
 }}
class  Current  extends  Account {
       float  deposit ;
       public void  deposit () {
        System. out.println ( " Enter  amount to be
             deposited");
        amount = amount + deposit;
 }
   public void  checkbal () {
        if (amount < minbal) {
        System . out. println ( " Insufficient  balance
        amount = amount - 500;
 }}
public void  withdraw () {
       float  withdraw ;
    if (withdraw > amount )
System . out . println ( " Insufficient  balance ");
    else
   amount = amount - withdraw ;

}
 }
```

```
class  Savings  extends  Account {
    float  deposit;
  public  void  deposit () {
      System out println (''Enter  deposit amt '')
         deposit=  sc. nextFloat ();
     }
   public void  interest() {
      double  interest ;
      interest =  ( 0.06 * amount).
      amount =  amount + deposit,
  }
~~public  void  interest ()~~
System.out.println ('' Amount  =  '' + amount);
 }
   public  void  withdraw () {
  System. out. println (''Enter  withdraw amount).
    if (withdraw > amount)
  System. out . println ( ''Insufficient  balance'');
    else
    amount = amount  - withdraw;
} }
public class Bank {
 public static  void  main ( String [] args) {
          int ch, i = 0;
          String type ;
   Current  c  =  new  Current ();
   Savings  S  =  new  Savings ();
 if (type = = "c'') {
     c.info();
     c. checkbal();
    while (i != 4) {
  System.out. Println (''Enter  your choice \n 1.
                                                 4 checkbal
  Display \n 2. Deposit \n 3. Withdraw \n 5 Exit'');
   choice = s. nextInt ();
   switch (choice) {
    case 1  : c1. deposit ();
              c1.display ();
             break;
case 2 : c1. withdraw ();
         c1. display ();
         break;
case 3 : c1. display ();
         break;
case 4 : c1. cheque ();
         break;
case 5 : System. exit (0);
         break;
default : System. out. println (" Invalid input!")
         break;
} } break;
case 2: type = "Savings";
while (x==0) {
    switch (choice) {
        case 1 : s1. deposit ();
                 s1. display ();
                 break;
        case 2 : s1. withdraw ();
                 s1. display ();
                 break;
        case 3 : s1. display ()
                 break;
        case 4 : s1. interest ();
                 s1. display ();
                 break;
        case 5 : System. exit (0);
    default : System. out. println (" Invalid input!")
              break;
    } } ~~bre~~ s. close ();
} }
```

Week 7

```
impot java.util.*;
class WrongAge Exception {
  String msg = new String();
  WrongAge (String x)
  { msg=x; }
public String toString () {
  return "Exception handled succesfully" +msg;
  } }
class father {
 int f_age;
 father() throws WrongAge {
 System.out.println ("Enter father's age");
  f_age = sc.nextInt();
  if (f_age<0)
  throw new WrongAge ("Father's age <=0"); }
  }
 }
 class son extends father {
  int s_age;
 son () throws WrongAge {
  System.out.println ("Enter son's age")
   s_age = s.nextInt();
  if (s_age <=0)
  throw new WrongAge("Son age > father age");
  }
 }
 class Main {
   public static void main (String[] args){
     try {
          son s = new son();
            s.display();
          }
        catch (WrongAge w){
             System.out.println(w);
        }
```

Week 8

```
import java.io.*;
import java.lang.Thread;
class NewThread implements Runnable{
 Thread t;
 NewThread(){
  t = new Thread(this,"Demo Thread");
  t.start();
 }
public void run(){
 try{
  while(true){
   System.out.println("CSE");
   Thread.sleep(2000);
  }
 }catch(Interupted exception e){
   }
 }
 }
 class ThreadDemo{
 public static void main(String args[]){
  new NewThread();
  try{
  while(true){
  System.out.println("BMS College of Engineering");
  Thread.sleep(10000);
  }
  }catch(Interrupted exception e){
    System.out.println("Main thread interupted");
   }
  }
 }
```

Output