

11/12/23

Lab-1

①

P1: Print hello world

Class World
{

{ Public Static Void Main (String ar[])

{ System.out.print ("Hello World");
}

OUTPUT

Hello World

P2: Basic Calculator

Class Calci

{

{ Public Static Void main (String ar[])

{ int a=10;

int b = 5 ;

System.out.println(a+b);

System.out.println(a-b);

System.out.println(a*b);

System.out.println(a/b);

}

}

OUTPUT

15

5

50

2.

P3: Fibonacci Sequence

Class Fib

{

Public static void main (String ar[])

{

int fib1=0;

int fib2=1;

System.out.print (fib1)

System.out.print (fib2)

-for (int i=0; i<=5; i++)

{

int fib3=fib1+fib2;

System.out.println (fib3);

fib1=fib2;

fib2=fib3;

}

}

OUTPUT

0

1

1

2

3

5

8

13

P4 P5: Check prime

(3)

```
class Prime
{
```

```
    public static void main (String ar[])
    {
```

```
        int a=17;
```

```
        if (a%2==0 || a%3==0 || a%5==0 || a%7==0)
        {
```

```
            System.out.print(" a is not a prime number");
        }
```

```
    else
```

```
        System.out.print (" a is prime no")
```

```
}
```

```
}
```

OUTPUT

a is a prime no.

201
10/10/23

LBB-02

Grocery Amount Calculator

Program:

```
import java.util.Scanner;
```

```
class Grocery,
```

{

```
    double qty_Dal, qty_pulses, qty_Sugar;
```

```
Grocery()
```

{

```
    qty_Dal = 1;
```

```
    qty_Pulses = 1;
```

```
    qty_Sugar = 0.5;
```

}

```
Grocery(double Some)
```

{

```
    qty_Dal = qty_Pulses = qty_Sugar = Some;
```

}

```
Grocery (double a, double b, double c)
```

{

```
    qty_Dal = a;
```

```
    qty_pulses = b;
```

```
    qty_Sugar = c;
```

}

```
Grocery (Grocery obj)
```

```
{ qty_Dal = obj.qty_Dal;
```

```
    qty_pulses = obj.qty_pulses;
```

```
    qty_Sugar = obj.qty_Sugar;
```

}

void calcAmount()

{

System.out.println("Total amount: ");

System.out.println(" " + qty.Dal * 150 + qty.Pulses * 80
+ qty.Sugar * 50);

}

}

Class Main

{

public static void main (String [] args)

{

Scanner input = new Scanner (System.in);

System.out.print ("Enter Dal quantity: ");

double dal = input.nextDouble();

System.out.print ("Enter pulses quantity: ");

double pulses = input.nextDouble();

System.out.print ("Enter Sugar quantity: ");

double Sugar = input.nextDouble();

Grocery g1 = new Grocery();

Grocery g2 = new Grocery(dal);

Grocery g3 = new grocery (dal, pulses, sugar);

Grocery g4 = new grocery (g3);

g1. calcAmount();

g2. calcAmount();

g3. calcAmount();

g4. calcAmount();

}

}

LAB - 03

```
import java.util.Scanner;
```

```
class Quadratic {
```

```
    int a, b, c;
```

```
    double r1, r2, d;
```

```
    void input() {
```

```
        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 / (2.0 * a);
```

```
            System.out.println("Roots are real and equal");
```

```
            System.out.println("Root 1 = Root 2 = " + r1);
```

```
}
```

```
        else if (d > 0) {
```

```
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
```

```
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
```

```

System.out.println("Roots are real and diff")
System.out.println("Root1 = " + s1 + ", Root2 = "
}

```

else if ($d < 0$) {

System.out.println ("Roots are imaginary")

$$s1 = -b / (2.0 * a)$$

$$s2 = \text{Math.sqrt}(\text{Math.abs}(d)) / (2.0 * a)$$

System.out.println ("Root3 = " + s1 + " + i" + s2)

System.out.println ("Root4 = " + s1 + " - i" + s2)

}

}

class QuadraticMain {

public static void main (String [] args)

Quadratic q = new Quadratic ();

q.input();

q.compute();

}

.

OUTPUT

Enter the co-efficients of a, b, c

1

2

1

Roots are real and equal

Root1 = Root2 = -1.0

Enter the co-efficient of a, b, c

6

8

0

Roots are real and distinct

$$\text{Root 1} = 0.0, \quad \text{Root 2} = -1.333$$

Enter the co-efficients of a, b, c

2

3

4

Roots are imaginary

$$\text{Root 1} = -0.75 + i 1.1989$$

$$\text{Root 2} = -0.75 - i 1.1989$$

(3)

LAB-04

```
import java.util.Scanner;
```

```
class Student
```

```
{ String usn, name;
  int n;
  int l[], m[];}
```

```
void acc()
```

```
{
```

```
Scanner s1 = new Scanner (System.in);
```

```
System.out.println ("Enter the USN of the student");
```

```
usn = s1.nextLine();
```

```
System.out.print ("Enter the name of the
student : ");
```

```
name = s1.nextLine();
```

```
System.out.print ("Enter the number of subjects");
```

```
n = s1.nextInt();
```

```
l = new int[n];
```

```
m = new int[n];
```

```
for (int i=0; i<n; i++)
```

```
{
```

```
System.out.print ("Enter the number of credits
of subject " + (i+1) + ": ");
```

```
l[i] = s1.nextInt();
```

```
System.out.print ("Enter the marks of subject "
+ (i+1) + ": ");
```

```
m[i] = s1.nextInt();
```

```
}
```

```
}
```

```
int gpa (int marks)
```

```
{
```

```
if (marks >= 90)
```

```
return 10;
```

```

else if (marks >= 80)
    return 9;
else if (marks >= 70)
    return 8;
else if (marks >= 60) 0
    return 7;
else if (marks >= 50)
    return 6;
else
    return 0;
}

```

~~double avg()~~

```

int tc = 0, sum = 0;
for (int i = 0; i < n; i++)
{
    tc = tc + c[i];
    sum = sum + gp(m[i]);
}

```

~~return sum / tc;~~

~~void disp()~~

```

System.out.println("Roll - " + roll);
System.out.println("Name - " + name);
}

```

~~class Main~~

{

~~public static void main (String args [])~~

```

Student s = new Student(1,
    "Sachin");
}

```

10/10/21

s.displ();

System.out.println("The score = " + s.getScore());



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

```
import java.util.Scanner;
```

```
abstract class Shape
```

```
{
    double x, y;
    double x, y;
    Shape(double x, double y)
    {
        this.x = x;
        this.y = y;
    }
}
```

```
abstract void PrintArea();
```

```
class Rectangle extends Shape
```

```
{
    Rectangle(double x, double y)
    {
        super(x, y);
    }
    void printArea()
    {
    }
}
```

```
System.out.println("Area of Rectangle : " + (x * y));
```

```
}
```

```
.
```

Class Triangle extends Shape

{

Triangle (double x, double y)

{ super (x, y);
}

void printArea ()

{

System.out.println ("Area of Triangle : " + (0.5 * x * y) +

" Square units")

}

}

Class Circle extends Shape

{

Circle (double x)

{ super (x, 0); } ✓

void printArea ()

{

System.out.println ("Area of Circle : " + (3.14 * x * x) +

" Square units")

}

Class ShapeRun

{ public static void main (String[] args) {

double x, y;

Scanner input = new Scanner (System.in);

System.out.println ("Enter length and width of

x = input.nextDouble();

y = input.nextDouble();

Rectangle");

Rectangle rectangle = new Rectangle(2, 3);
rectangle.drawRect();

System.out.println("Enter height and base of Triangle");
 $x = \text{input.nextInt}();$
 $y = \text{input.nextInt}();$

Triangle triangle = new Triangle(x, y);
triangle.printArea();

System.out.println("Enter radius of Circle");

$x = \text{input.nextInt}();$

Circle circle = new Circle(x);

circle.printArea();

input.close();

}

}

OUTPUT

Enter length and width of Rectangle:

10

20

Area of Rectangle : 200.00 square unit

Enter height and base of Triangle:

10

20

Area of Triangle : 100.00 square unit

Enter radius of circle

10

Area of Circle: 314.0 Square unit

Lab-06

Create a class book that contains four members name, author price, and num pages. Include a constructor to set and get the details of the object. Include a `toString()` method that could display the complete details of the book.

Develop a java program to create n book objects

```
import java.util.Scanner;
```

```
class Books
```

```
{
```

```
    String name, authr;
```

```
    int price, num_page;
```

```
Books(String name, String authr, int price, int num_page)
```

```
{ this.name = name;
```

```
    this.authr = authr;
```

```
    this.price = price;
```

```
    this.num_page = num_page;
```

```
}
```

```
Scanner input = new Scanner(System.in);
```

```
Books() {}
```

```
void accept()
```

```
{
```

```
    System.out.print("Enter name of book: ");
    name = input.nextLine();
```

```
    S.O.P("Enter name of author: ");
```

```
    author = input.nextLine();
```

```
    S.O.P("Enter price of the book: ");
    price = input.nextInt();
```

S.O.P ("Enter no of pages in the book :"),
 num-pages input.nextInt();
 S.O.P ("Yn");
 "y"

public String toString()

{
 String name, author, price, num-pages;
 name = "Book name : " + this.name + "\n";
 author = "Author name : " + this.author + "\n";
 price = "Price : " + this.price + " BDT";
 num-pages = "Number of pages : " + this.num-pages + "
 pages\n";

return name + author + price + num-pages;

}
 "y"

class BookRun

{

Public static void main (String[] args){
 Scanner input = new Scanner (System.in);

S.O.P ("Enter number of books : ");
 int n = input.nextInt();
 S.O.P ("Yn");

Books [] book = new Books [n];

for (int i=0; i<n; i++)
 {
 S.O.P ("Book " + (i+1) + ": ");
 book [i] = new Books ();
 book [i].accept();

(14)

```

for (int i=0; i<n; i++)
{
    S.O.P ("Book", (i+1) + ": " + book[i] + "\n");
}
input.close();
}

```

OUTPUT

Enter number of books: 2.

Book 1:

Enter name of book: abc

Enter name of author: ef

Enter price of the book: 1000

Enter no of pages in the book: 200

Book 2:

Enter name of book: def

Enter name of author: ghe

Enter price of the book: 2000

Enter no of pages in the book: 23

Book 1:

Book name: abc

Author name: ef

Price: 1000 P.

Number of pages: 200 page

Book 2:

Book name: def

Author name: ghe

Price: 2000 P.

Number of pages: 23 page

~~Right~~
8/10/2014

Lab-07

(18)

Bank programme

import java.util.Scanner;

class Account

```
{ String customerName;
  long accountNumber;
  String accountType;
  double balance;
```

```
public Account (String customerName, long accountNumber,
                String accountType, double balance)
```

{

```
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;
```

}

public void deposit (double amount)

{

balance += amount;

System.out.println ("Deposit successful. Updated balance: " + balance);

}

public void displayBalance ()

~~```
{ System.out.println ("Account Number: " + accountNumber);
 System.out.println ("Customer Name: " + customerName);
 System.out.println ("Account Type: " + accountType);
 System.out.println ("Balance: " + balance);
```~~

}

}

James Shrock - Extent Seamed

1

Pubic Lacer (sliding obturator, long narrow double blades)

Superintendente, alcaldesas, "Jang", y

四

Refer to separate Report Interest (doubt not)

四

Death interest = balance \* state / 100.

#### Other institutions

S.C.R. C. Indian competitor and distributor

Updated balance below:

Public word withdrawal (double meaning)

10

3) Current & history

3

*belarus = : amarillo:*

S.C.P. in ("Withdrawn Successful Update" block  
before)

Check Minimum Balance:

10

24

2. S. S. P. ("Sneffles goes" withdrawals)

1

Franklin Glass Bass

2 public stocks were made (Schoell 1997).

12. Second class plain cotton 100% white

System and parts ('Ente existente nascendo per suu  
naturam');

②

```

long SAN = SI.nextInt();
S.O.P ("Enter account number for Savings Account : ");
long SBN = SI.nextInt();
S.I.O.P ("Enter initial balance for Savings Account : ");
double SIB = SI.nextDouble();
SA = new Savings (SCN, SAN, SIB);

```

```

S.O.P ("Enter customer name for Current Account : ");
String CCN = SI.next();
S.O.P ("Enter account number for Current Account : ");
long CAN = SI.nextInt();
S.O.P ("Enter initial balance for Current Account : ");
double CIB = SI.nextDouble();
S.O.P ("Enter minimum balance for Current Account : ");

```

double MB = SI.nextDouble();

S.O.P ("Enter service charge for current Account : ");

double SC = SI.nextDouble();

CurrAcnt CA = new CurrAcnt (CCN, CAN, CIB, MB, SC);

S.O.P ("Enter deposit amount for Savings Account : ");

double SDA = SI.nextDouble();

SA.deposit (SDA);

S.O.P ("Enter interest rate for Savings Account : ");

double SSR = SI.nextDouble();

SA.compute AND Deposit Interest (SSR);

S.O.P ("Enter withdrawal amount for Savings Account : ");

double SWR = SI.nextDouble();

CA.deposit (SWR);

S.O.P ("Enter withdrawl amount for Current Account : ");

Double CWA = S.nextBalance();

CWA.withdraw(CWA);

S.O.P ("Enter Ending Balance : ");

S.O.P (" Savings Account ");

S.A.displayBalance();

S.O.P("In Current Account");

In. depositBalance();

}

### Output:

Savings Account:

Enter customer name: Dharmik

Enter account number: 1234

Enter initial balance: 1000

Enter deposit amount for savings account: 30

Enter interest rate for Savings Account: 5

Enter withdrawal amount for savings acc: 200

Current Account:

Enter customer name: Phansen

Enter account number: 5678

Enter balance: 2000

Enter minimum balance: 1000

Enter annual charge: 10

Enter deposit amount for current Account: 500

Enter withdrawal amount for current Account: 1000

Final Balance:

Savings Account: ~~₹ 1305.0~~

Account Number: 123456

Customer Name: Dhanush

Account type: Savings

Balance: ₹ 1305.0

Current Account:

Account Number: 54321

Customer Name: Dhanush.C.

Account Type: Current

Balance: ₹ 820

LAB-08

Package

package CIF;

public class Student {

public String usn, name;

public int Sem;

public Student(String usn, String name, int Sem) {

this.usn = usn;

this.name = name;

this.Sem = Sem;

}

3.

package CIF;

import CIF.Student;

public class Intern extends Student {

public double[] internalMarks = new double[5];

public Intern(String usn, String name, int Sem,

double[] internalMarks) {

super(usn, name, Sem);

this.internalMarks = internalMarks;

}

}

package SEE

import CIF.\*;

public class External extends Student {

public double[] SeeMarks; new double[s];  
 public External (String usn, String name, int sem, double[] marks);  
 Super (usn, name, sem);  
 this. SeeMarks = SeeMarks;  
 }

y

```
import CIF.Interfaces;
import SEE.External;
import java.util.Scanner;
```

```
public class FinalMarks{
```

```
public static void main (String [] args) {
```

```
Scanner input = new Scanner (System.in);
```

```
System.out.print ("Enter the number of Students: ");
```

```
int n = input.nextInt();
```

```
input.nextLine();
```

```
Interfaces[] Internals = new Interfaces [n];
```

```
External[] externals = new External [n];
```

```
for (int i=0; i<n; i++) {
```

```
S.O.P ("Enter details for Internal Marks of");
```

```
Student " + (i+1));
```

```
System.out.print ("Enter USN: ");
```

```
String usn = input.next();
```

```
System.out.print ("Enter Name: ");
```

```
String name = input.next();
```

```

System.out.print("Enter Semester:");
int sem = input.nextInt();
input.nextLine();
double[] internalMarks = new double[5];
System.out.print("Enter internal marks for 5 courses:");
for (int j=0; j<5; j++) {
 internalMarks[j] = input.nextDouble();
}
internal[i] = new Internal(usr, name, sem, internalMarks);
}

```

```

for (int i=0; i<n; i++) {
 System.out.print("Enter details for SEE mark
of Student " + (i+1));
 String usn = internal[i].usn;
 String name = internal[i].name;
 int sem = internal[i].sem;
 double[] seeMarks = new double[5];
 System.out.print("Enter SEE Marks for 5 courses:");
 for (int j=0; j<5; j++) {
 seeMarks[j] = input.nextDouble();
 }
 external[i] = new External(usr, name, sem, seeMarks);
}

```

```

System.out.print("In Final Marks of Students:");
for (int i=0; i<n; i++) {
 System.out.print("Student " + (i+1) + ": USN: " +
 internal[i].usn + " Name: " +
 internal[i].name + " In Semest: " + internal[i].sem);
}

```

```
System.out.print("Total Marks : ");
for(int j=0; j<5; j++) {
```

```
 System.out.print("Subject " + (j+1) + ":" +
 (int)internalMarks[j] / 2) + (int)externalMarks[j],
 + "\n");
```

3

```
System.out.println();
```

}

{

}

### OUTPUT

Enter the number of Students: 1.

Enter details of Student 1

Enter USN: 085

Enter name: Dhanush

Enter Semester = 3

Enter Internal marks for 5 course: 80

80

80

85

100

Enter S.E.E marks for 5 courses of Student 1

80

80

90

85

100.

~~85 90 80 80 100~~

Final Marks of Student:

Student 1:

USN: 085

Name: Dhanusha C

Semester: 3

Subject 1 : 80

Subject 2 : 85

Subject 3 : 85

Subject 4 : 90

Subject 5 : 100

✓ 87/100

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, implement a constructor which takes the age and throws the exception WrongAge() when the input age is 0. In Son class implement a constructor that uses both father and son's age and throws an exception if son's age is greater than or equal to father's age.

import java.util.Scanner;

class WrongAge extends Exception

{}

    Public WrongAge( String str)

    {

        Super(str)

    }

}

class Father

{

    int fatherAge;

    Father (int fatherAge) throws WrongAge

{

    this.fatherAge = fatherAge;

    if (fatherAge < 0).

        throw new WrongAge ("Father's age cannot be less than Zero");

}

}

class Son extends Father

{

int sonAge;

Son (int fatherAge, int sonAge) throws WrongAge

{

super(fatherAge);

this.sonAge = sonAge;

if (sonAge < 0)

throw new WrongAge ("Son's age cannot be less than zero").

if (sonAge >= fatherAge)

throw new WrongAge ("Son's age cannot be greater than or equal to Father's age").

}

}

class AgeExceptionMain

{

public static void main (String [] args)

{

Scanner input = new Scanner (System.in);

System.out.print ("Enter father's age: ");

int fatherAge = input.nextInt();

System.out.print ("Enter son's age: ");

int sonAge = input.nextInt();

try {

Son son = new Son (fatherAge, sonAge);

System.out.println ("Father's and Son's age are value");

3. catch (WrongAge e):

{

System.out.println ("Exception : "+e);

}

}

.

### Output:

Enter father's age:-50

Enter son's age :-20

Father's and son's age are valid

Enter father's age:-50

Enter son's age :60.

Exception : WrongAge : Son's age cannot be greater than or Equal to Father's age

Enter father's age:-1

Enter son's age :20

Exception : WrongAge : Father's age cannot be less than zero

Enter father's age:0

Enter son's age :-1

Exception : WrongAge : Son's Age cannot be less than zero

Write a program which creates two threads  
 one thread displaying "BMS College of Engineering"  
 once every 10 seconds and another displays  
 "CSE" once every two seconds

```
class BMSSdisplay extends Thread
```

```
{
```

```
public void run()
```

```
{
```

```
for (int i=0; i<3; i++)
```

```
{
```

```
System.out.println ("BMS college of Engineering");
```

```
try {
```

```
Thread.sleep(10000);
```

```
}
```

```
catch (InterruptedException e) {
```

```
System.out.println ("Exception : "+e);
```

```
}
```

```
}
```

```
class CSEdisplay extends Thread
```

```
{
```

```
public void run()
```

```
{
```

```
for (int i=0; i<10; i++)
```

```
{
```

```
System.out.println ("CSE");
```

```
try {
```

```
Thread.sleep (2000);
```

```
}
```

```
catch (InterruptedException e) {
```

```
{}
```

System.out.println("Exceptions : " + e);

{

}

{

}

class BMSThread.

{

public static void main(String[] args)

BMSdisplay bms = new BMSdisplay();

CSEdisplay cse = new CSEdisplay();

bms.start();

cse.start();

{

}

### OUTPUT

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

BMS

College of Engineering

start  
25/10/2024

(33)

## AWT

Create label, button and textField in a frame using AWT.

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

```
public class AwtExample extends WindowAdapter {
```

```
Frame f;
```

```
AwtExample() {
```

```
f = new Frame("Employee id:");
```

```
f.addWindowListener(this);
```

```
Label l = new Label("Employee id:");
```

```
Button b = new button("Submit");
```

```
TextField t = new TextField();
```

a. SetBounds (20, 80, 80, 30);

b. SetBounds (20, 100, 80, 30);

c. SetBounds (100, 100, 80, 30);

```
f.add(b);
```

```
f.add(l);
```

```
f.add(t);
```

```
f.setSize(400, 300);
```

```
f.setTitle("Employee info");
```

```
f.setLayout(null);
```

```
f.setVisible(true);
```

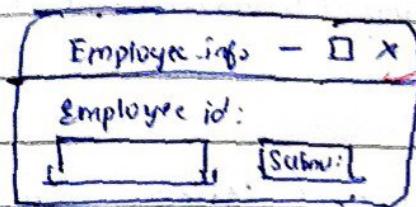
3.

```
public void windowClosing (WindowEvent e) {
 System.exit(0);
}
```

```
public static void main (String [] args) {
 AWTExample aut obj = new AWTExample ();
}
```

}

### OUTPUT



2. Create a button and add a action listener for Mouse click

```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowAdapter
implements ActionListener {
 Frame f;
 Textfield tf;
 EventHandling () {
```

```
 f = new Frame ();
 f.addWindowListener (this);
 tf = new TextField ();
 tf.setBounds (60, 50, 150, 20);
 Button b = new Button ("click me");
 b.setBounds (100, 120, 80, 30);
 b.addActionListener (this);
```

```

 f.add(lb); f.add(tf);
 f.setSize(300, 300);
 f.setLayout(null);
 f.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
 tf.setText("Welcome");
}

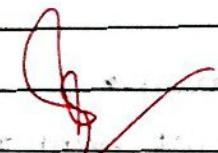
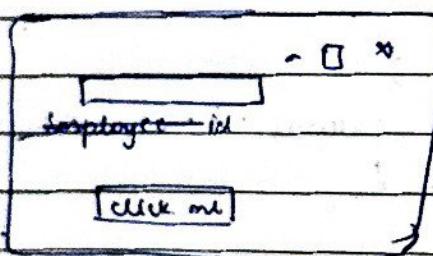
```

```

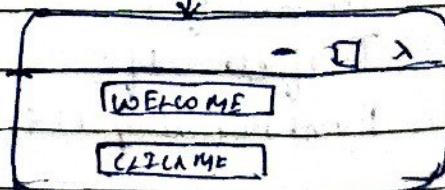
public void windowClosing(WindowEvent e) {
 System.exit(0);
}

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

```

OUTPUT

→ After clicking, click me



## Programs on IO

1. Example 1

```

import java.io.*;
public class ByteArrayInput {
 public static void main(String [] args)
 throws IOException {
 }
}

```

```

byte[] buf = {35, 36, 37, 38};
ByteArrayInputStream byt = new ByteArrayInputStream(buf);
int k = 0;
while ((k = byt.read()) != -1) {
 char ch = (char) k;
 System.out.println("ASCII value of character is " + k + "; Special character is " + ch);
}

```

OUTPUT

|            |                         |
|------------|-------------------------|
| ASCII : 35 | Special character : #   |
| ASCII : 36 | Special character : \$. |
| ASCII : 37 | Special character : %.  |
| ASCII : 38 | Special character : &   |

Example-2.

```

public class FileEx {
 public static void main(String ar[]) throws
 IOException {
 FileInputStream fin = new FileInputStream("Example1");
 int content;
 System.out.println("Remaining bytes that can be
 read : " + fin.available());
 content = fin.read();
 System.out.print((char) content + " ");
 System.out.print(content + " ");
 System.out.println("Remaining bytes that can
 be read : " + fin.available());
 System.out.println("Remaining bytes that can be
 read : " + fin.available());
 }
}

```

4

3.

(S) 24