Date __/__/__
Page _____

Q1. Develop a java program to create a class with members usn, name, an array credits and array marks include methods to accept and display details and a methode to calculate SGPA of a student.

=> import java.util.*;
class student{
    string usn;
    string name;
    int credits[];
    int marks[];
}
Void accept details(){
    Scanner s = new Scanner(system.in);
    System.out.println("Enter the usn :");
    usn = s.next();
    s.nextline();
    System.out.println("Enter the Name :");
    name = s.next();
    s.nextline();
    System.out.println("Enter the no: of subjects");
    int n = s.nextInt();
    credits = new int[n];
    marks = new int[n];
    s.nextline();
for(int i=0; i<n; i++){
    System.out.println("Enter the credits for subject" (i+1) + " ");
credits[i] = s.next Int();
System.Out.println("Enter the marks for subject" + (i+1) + " ");

```
        marks[i] = s.nextInt();
    }
}
Void display details(){
System.out.println("usn "+usn);
System.out.println("Name "+ name);
for(int i=0; i<credits.length; i++){
    System.out.println("subjects"+(i+1)+" "+
        credits[i]);
}
}
double calculate SGPA(){
float sum=0;
for(int i=0; i<5; i++){
    if(marks[i]>=90){
        Sum = Sum + (credits[i]*10);}
else if(marks[i]>=80){
    Sum = Sum + (credits[i]*9);
}
else if(mark[i]>=70){
    Sum = Sum + (credits[i]*8);
}
else if(mark[i]>=60){
    Sum = Sum + credits[i]*7);
}
else if(mark[i]>=50){
    Sum = Sum + credits[i]*6);
}
else if(mark[i]>=35){
    Sum = Sum + (credits[i]*5);
}
else{
    Sum = Sum + (credits[i]*0);
}
```

```
int cresum=0;
for(int i=0; i<5; i++){
    cresum += credits[i];
}
float sgpa = sum/(float)cresum;
return sgpa;
}
}
Public class studentsgpa{
public static void main(string args[]){
Student s = new student();
s.accept details();
display details();
System.out.println("SGPA "+s.calculate
    SGPA());
}
}
```

Output :

=> Java.c Student sgpa.java
=> java studentsgpa
Enter the usn : 1BM22CS164
Enter the Name : Aarohi
Enter the no of subjects : 5
Enter the

Q. Develop a java program that prints all real solutions to the Quadratic equation $ax^2+bx+c=0$. read in a, b, c and use the Quadratic formula.

```
import java.util.Scanner;
class First {
    Public static Void main (string args[]){
    int a, b, c;
    double r1, r2, d;
    System.Out.println("enter the coefficient
        of Quadratic equation:");
    Scanner s1 = new Scanner(System.in);
    a = s1.nextInt();
    b = s1.nextInt();
    c = s1.nextInt();
    if (a==0){
        System.Out.println("coefficients are invalid:")
    }
    else
        d = b*b-(4*a*c)
        if (d>0){
            System.Out.println("It has real and distinct
                roots");
        r1 = ((-b+math.sqrt(d)))/(2*a);
        r2 = ((-b-Math.sqrt(d)))/(2*a);
    System.Out.println("The roots are "+r1+" and "
                +r2);
        }
    else if (d==0){
        System.Out.println("It has real and equal
                roots");
        r1 = (-b)/(2*a);
    System.Out.println("The roots are "+r1+" and "
+r1);
    }
    else{
        System.Out.println("It has no real roots");
            }
        }
    }
```

* Output

```
>javac First.java
>java First
Enter the coefficients of Quadratic equation
1
2
1
It has real and equal roots
The roots are -1.0 and -1.0
>java first
Enter the Coefficients of Quadratic equation
1
1
4
It has no real roots
>java first
Enter the Coefficient of Quadratic equation
1
4
1
It has real and distinct roots
The roots are -0.2679491924311228 and
-3.7320508075688772
```

*) Book Program

```
Public class Book {
    string name;
    string author;
    double price;
    int numpages;
Public Book (string name, string author,
  double price, int numpages){
this.name = name;
this.author = author;
this.price = price;
this.numpages = numpages;
}
  Public Void setDetails (string name, string
    author, double Price, int numpages){
   this.name = name;
    this.author = author;
    this.price = price;
    this.numpages = numpages;
}
Public string tostring(){
    return Book Details: \n Name: "+name+
"\n Author:" + author + "\n Price: " + Price +
"\n Number of pages:" +numPages;
}
Public static Void main(string[] args){
    Scanner s = new Scanner();
    int n;
    System.Out.println("enter the no: of books:");
n=s.nextInt();
  Book[] books = new Book[n];
for(int i=0; i<n; i++){
    System.Out.println(books[i].tostring());
    System.Out.println("---------");
}
}
}
```

Output:-

```
Enter the Number of books.
3
Book Details:
Name: Book 1
Author: Author 1
Price: 200.0
Numbers of Pages: 100
----------
Book Details:
Name: Book 2
Author: Author 2
Price: 205.0
Number of Pages: 150
----------
Book details:
Name: Book 3
Author: Author 3
Price: 3000.0
Number of Pages: 200.
----------
```

* Develop a Java program to create an abstract class named shape that contains two integers and an empty method named 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 one methode PrintArea() that Prints the area of the given shape.

→

```
abstract class shape{
    int d1;
    int d2;
Public shape (int d1, int d2){
    this.d1 = d1;
    this.d2 = d2;
}
Public abstract void PrintArea();
}
class rectangle extends shape{
    Public rectangle(int length, int width){
    super (length, width);
}
  Public Void PrintArea(){
    int area = d1 * d2;
    System.Out.println("rectangle Area" + area);
    }
  }
  class triangle extends shape{
  Public Triangle (int base, int height){
     super (base, height);
  }
  Public Void PrintArea(){
     double area = 0.5 * d1 * d2;
  System.Out.println("Triangle Area:" + area);
    }
  }
class circle extends shape{
     Public circle (int radius){
        super (radius, 0);
  }
  Public Void PrintArea(){
     double area= Math.PI * d1 * d2;
  System.Out.println(" Circle Area:" + area);
    }
  }
Public class shape Main {
   Public static void main(String[] args
{
Rectangle r=new Rectangle(5,10);
   r.printArea();
  Triangle t= new Triangle(3,6);
  t.printArea();

  Circle c= new Circle(4);
  c.printArea();
  }
}
```

* ~~CIE, SEE~~
* CIE, SEE package programs

student.java

```
Package CIE;
Public class student {
  string USN;
  string name;
  int sem;
Public student (string usn, string name,
  int sem) {
  this.usn = usn;
  this.name = name;
  this.sem = sem;
  }
}
```

Import Java.util.Scanner;

```
class wrongage extends Exception{
    Public string toString(){
    return "age cannot be less than 0";
    }
}
class cannotbe extends Exception {
    Public string toString(){
        return "Son cannot be older than father"
    }
}
class Father {
    int age;
  Father(int age) throws wrongage {
    this.age = age;
    if(age<0) {
        throw new wrongage();
        }
    }
}
class son extends Father {
    int Fatherage, sonage;
  Son(int Fatherage, int sonage) throws wrongage,
      cannotbe {
  Super(sonage);
  this.Fatherage = Fatherage;
  this.sonage = sonage;
  if (son age > Fatherage){
      throw new cannotbe();
  } else {
      System.Out.println("father age" + Fatherage +
      "\n son age = " + Sonage);
      }
  }
}
class Main {
    Public static void main(String[] args) {
    int fatherage, sonage;
  Scanner s = new scanner(system.in);

  try {
      System.Out.println("enter Father age:");
    father age = s.nextInt();
    Father f = new Father(father age).
  System.Out.println("enter son age:");
    Son age = s.next Int();
    Son Sonobject = new son(fatherage, sonage);
  }
```

```
catch (WrongAge | CannotBe e) {
    System.Out.println("Exception:" +e);
  }
 }
}
```

O/P

Enter father age:
45
Enter son age: -2
exception: age cannot be less than 0.

```
Package CIE
public class student {
        public string usn;
        public string name;
        public int sem;

    public student(string usn, string name,
            int sem ){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
Package CIE
public class Internals {
    Public int[] marks = new int[5];
        Public Internals (int [] marks ){
            if (marks.length == 5) {
                this.marks = marks;
            }else {
                System.out.println("please provide
                marks for all five courses");
            }
        }
    }
    Package SEE
    import CIE.students;
public class externals extends students {
    Public int[] marks = new int[5];
    Public external (string usn, string name, int sem,
            int marks[] ){
```

```
        super(usn, name, sem);
        if( marks.length == 5 ){
            this.marks = marks;
        }
    else {
        System.Out.println("please provide
            marks for all courses");
        }
    }
}
CIE.student;
CIE.internals;
SEE.externals;
class final marks {
        public static void main(String[] args)
    student [] students = new student [2];
    int[] ...
    int [] studentInternalmarks = {80, 75, 85, 90, 78};
int [] student SEE Marks = {70, 65, 75, 80, 60};
students[0] = new external ("1BM22CS001" "Alice",
                5, studentSEEMarks)
Internals studentInternals = new Internals(
        studentInternalmarks);
int[] student2 Internalmarks = {85, 80, 90, 88, 78};
int[] student2SEEmarks = {75, 70, 80, 85, 65};
student [1] = new external("1BM22CS002"
        "John", 5, studentSEEmarks);
Internals student2Internals = new Internals(student
        2 Internalmarks);
for (int i=0; i<students.length; i++) {
    int [] final marks = new int[5]
    System.out.println("student.:" + students[i].name);
    System.Out.println("usn :" + students[i].usn);
    System.out.println("semester:" + students[i].sem);

    for(int j=0; j<5; j++){
        finalmarks[j] = studentInternals.marks[j]
            + students[i].marks[j];
    System.Out.println("final marks =" + (j+1) + ":" +
            finalmarks[j]);
    }
    System.Out.println();
    }
}
```

```
class Generic <A, B, C, D, E >{
    A a;
    B b;
    C c;
    D d;
    E e;

Generic (A a1, B b1, C c1, D d1, E e1){
    a= a1;
    b= b1;
    c = c1;
    d = d1;
    e = e1;
}
Void show type ( ){
    System.Out.println (a. get class ( ). get Name()
                        (b. get class ( ) .get Name()
                        (c. get class ( ). get Name ( ).
                        (d. get class ( ). get Name ( ).
                        (e. get class ( ). get Name ( )
}
A return type ( ) {
    return a; }
B return type1( ) {
    return b;
}
C return type 2 ( ) {
    return c;
}
D return type3 ( ) {
    return d;
}
E return type 4 ( ) {
    return e;
}
}
class Generic demo {
    public static void main (string[ ] args) {
Generic <Integer, float, string, Double, Boolean>
                    g1;
g1 = new Generic <Integer, float, string, double,
                Boolean>
    ( 1 float 1.222 "BMS", 2.1113, true);
g1. showtype ( );
System. Out. println
int x = g1. return type ( );
System. Out. println (x);
float x1 = g1. return type ( );
System. Out. println (x1);
String x2 = g1. return type ( );
System. Out. println (x2);
Double x3 = g1. return type 3 ( );
System. Out. println (x3);
Boolean x4 = g1. return type 4 ( );
System. Out. println (x4);
    }
}
```

O/p :-

```
Java. lang. Integer
Java. lang. Float
Java. lang. string
Java. lang. double
Java. lang. Boolean
```

1

1. 2322

BMS

2. 11332

true.

Q) write a program to show user defined exception. create class employee use inheritance. throw exception when manager salary is less than employee.

```
class Myexception extends Exception {
    public string tostring() {
        return "manager salary cannot be
    less than worker".
    }
}
class Mangger {
        int manager salary;
        public Manager (int managerSalary){
         this.manager salary = manager Salary;
        }
    }
class Worker extends manager{
    int workerSalary;
    public worker (int worker salary, int Managersalary
        throws Myexception {
      super (manager Salary);
     if (manager salary < worker Salary ){
      throw new my exception ();
      }
    }
}
public class Employee1 {
    public static Void main (string[] args) {
     try {
       Worker W = new worker (40000, 30000);
    }
     Catch (Myexception e){
        System.Out.println (e); } } }
```

Q) Write a program which creates two threads one displaying "BMS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

=>
```
class displayThread extends Thread {
    private String message;
    private int interval;
    public DisplayThread(String message
                         int interval);
{
this.message = message;
this.interval = interval;
}
public void run() {
    while(true) {
        try {
        System.out.println(message);
        Thread.sleep(interval);
        }
    Catch (InterruptedException e) {
    e.printStackTrace();
        }
    }
}
class Display message {
    Public static Void main(String[] args){
    Thread bmsThread = new DisplayThread
    ("BMS College", 10000);
    Thread cseThread = new DisplayThread
            ("cse", 2000);
    bmsThread.start();
    cseThread.start();
    }
}
```

O/p

BMS college
CSE
CSE
CSE
CSE
CSE
BMS college
CSE
CSE
CSE
CSE
CSE
BMS College.

develop a java program to create a class Bank that maintains two kind of accounts its customers, one called savings account and the other current account. the savings account provides compound interest and withdrawal facility have no checkbook facility The current account provides chequebook facility no interest current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

create a class account that stores customer name, account number and type of account from this derive the classes current-account and savings account to make them more specific to their requirements include the necessary methods in order to achieve the following tasks,

a) Accept deposit from customer & update the balance
b) display the balance
c) compute & deposit interest
d) permit withdraw & update the balance.
check for minimum balance impose penalty & update the balance.

```
import java.util.scanner;
class Bank {
    double accno;
    string name;
    string type;
    double balance;
Public Bank (double accno, string name, string type,
        double balance) {
    this.accno = accno;
    this.name = name;
    this.type = type;
    this.balance = balance;
  }
Public Void display () {
    System.out.println("Current balance is :"+
        balance);
    }
public Void deposit (double deposit) {
    this.balance += deposit;
  System.out.println("update balance is : "+balala
  }
public void withdraw (double amount) {
  if (amount > balance) {
  System.out.println("Insufficient balance");
  }else {
    balance -= amount;
    System.out.println("After withdrawing, the
        balance is: "+balance);
    }
  }
```

```
class savings extends Bank {
    double rate;
    int time;

public saving (double accno, string name
        double balance, int time, double rate) {
    super (accno, name, "savings", balance);
    this.time = time;
    this.rate = rate;
    }
public Void CalculateInterest () {
    balance += (balance * time * rate) / 100;
    }
}
class current extends Bank {
    double minBalance;
    public current (double accno, string name, double
    balance, double minBalance) {
    super (accno, name, "current", balance);
    this.minBalance = minBalance;
    }
public Void applyServiceCharge () {
    if (balance < minBalance) {
    System.Out.println ("service charge of 5% is
        applied:");
    balance -= balance * 0.05;
    }
}
```

AWT

```
import java.awt.*;
import java.awt.event.*
public class DivisionMain extends Frame implements
Action listener
{
    TextField num1, num2;
    button dresult label Out Rout;
    string Out = " ";
    double resultNum;
    int flag = 0;
    public Division Main ()
{
Set layout (New Flowlayout());
d Result =new Button("result");
Label number1 =new Label("Number :1", label RIGHT)
Label number2 = New label("Number :2", label RIGHT)

num1 =new Textfield (5);
num 2 = new Textfield(5);
Out Result = new label ("Result:", label.RIGHT);
add (number 1);
add (num 1);
add(number 2);
add(num 2);
add(d Result);
add( out Result);
num1.add Action listener(this);
num2. add Action listener (this);
d. result . add Action listener (this);
add window listener(new window Adapter()
    {
    public Void window closing (window Event We)
{ system. exit (0);
        }
    );
public void Action performed (ActionEvent ae)
{
double n1, n2;
try{
if(ae.getsource() == d Result){
    n1 = Double.parse Double (num1. getText());
n2= Double .parseDouble (num2. get Text());
Out = n1 + "  "+n2;
resalt Num = n1/n2;
Out += string.value of (result Num);
repaint();
}
Catch (Arithmatic Exception (e)
{
flag = 1;
Out = "Divide By 0 Exception!" +e2;
repaint();
}
Catch (Number Format Exception e1)
{
flag = 1;
Out = "Number format Exception "+ e1;
repaint();
}
}
public void paint (Graphics g){
    if ( flag == 0)
    g. Draw String (out, Out Result.get x() +
        Out Result. get width (), out Result. get()+
```

```
out result.getHeight()-8);
else
    g.Drawstring (out,100,200);
}
public static void main(string args[]){
    Division main dm = new Division main();
    dm.setsize(new Division(200,400));
    dm.setTitle("Division of integers");
    dm.setVisible(true);
}
}
```

o/p

Number 1: [12] Number 2: [0]

[Result] Result: 12 0 0 infinity

Number 1: [12] Number 2: [a]

[Result] Result:

Number format Exception:

Java.Lang.NumberFormatException. for input string 'a'

Number 1: [0] Number 2: [12]

[Result] Result 0 012.00

Number 1: Number 2: 12

[Result] Result 12 0.5000