

1. Write a simple java application, to print the message, "Welcome to java"

```
public class Welcome {  
    public static void main(String[] args) {  
  
        System.out.print("Welcome to Java !!!");  
    }  
}
```

Output:

Welcome to Java !!!

2. Write a program to display the month of a year. Months of the year should be held in an array.

```
public class ArrayDemo {  
    public static void main(String[] args) {  
        int n;  
        String[] month = {"Jan", "Feb", "March", "April", "May", "June", "July", "August", "Sep", "Oct", "Nov", "Dec "};  
        for (int i = 0; i < 12; ++i) {  
            System.out.print(i+1 +". "+ month[i]);  
        }  
    }  
}
```

Output:

1. Jan
2. Feb
3. March
4. April
5. May
6. June
7. July
8. August
9. Sep
10. Oct
11. Nov
12. Dec

3. Write a java program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

```
public class FuncOver {  
    public void addNum() {  
        int a = 10;  
        int b = 20;  
        System.out.println("Sum = " + (a + b));  
    }  
    public void addNum(int a, int b) {  
        System.out.println("Sum = " + (a + b));  
    }  
    public void addNum(double a, double b) {  
        System.out.println("Sum = " + (a + b));  
    }  
}
```

```
public class MainFunc {  
    public static void main(String[] args) {  
        FuncOver n1 = new FuncOver();  
        System.out.println("No arguments");  
        n1.addNum();  
        System.out.println("integer arguments");  
        n1.addNum(5, 2);  
        System.out.println("Float arguments");  
        n1.addNum(5.2, 2.5);  
    }  
}
```

Output:

```
No arguments  
Sum = 30  
integer arguments  
Sum = 7  
Float arguments  
Sum = 7.7
```

4. Write a program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide. A main function should access the methods and perform the mathematical operations.

```
public class AddSub {  
  
    int a, b;  
  
    void add() {  
        int c = a + b;  
        System.out.println("Sum " + c);  
    }  
  
    void sub() {  
        int c = a - b;  
        System.out.println("Difference " + c);  
    }  
}
```

```
public class MulDiv extends AddSub {  
  
    void mul() {  
        int c = a * b;  
        System.out.println("Product " + c);  
    }  
  
    void div() {  
        int c = a / b;  
        System.out.println("Quotient " + c);  
    }  
}
```

```
public class MainClass {  
  
    public static void main(String[] args) {  
        MulDiv ob = new MulDiv();  
        ob.a = 20;  
        ob.b = 10;  
        ob.add();  
        ob.sub();  
    }  
}
```

```
        ob.mul();  
        ob.div();  
    }  
}
```

Output:

Sum 30
Difference 10
Product 200
Quotient 2

5. Write a program with a class variable that is available for all instances of a class. Use static variable declaration.

```
public class StaticDemo {

    int regno;
    String name;
    static int age;

    StaticDemo( int regno, String name,int age){

        this.regno = regno;
        this.name = name;
        this.age = age;
    }

    void display(){

        System.out.println("-----");
        System.out.println("Regno: " + regno);
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }

}



---



public class MainClass {

    public static void main(String args[]) {

        StaticDemo a = new StaticDemo(101, "Lalith", 20);
        StaticDemo b = new StaticDemo(102, "Deekshitha", 19);
        StaticDemo c = new StaticDemo(103, "Lakish", 18);

        a.display();
        b.display();
        c.display();
    }

}
```

Output:

Regno: 101

Name: Lalith

Age: 18

Regno: 102

Name: Deekshitha

Age: 18

Regno: 103

Name: Lakish

Age: 18

6. Write a java program to create a student class with following attributes: Enrollment_id: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition, write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.

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

```
public class StudInfo {
    int enrollId, sub1, sub2, sub3, total;
    String name;

    StudInfo() {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter enrollment Id");
        enrollId = in.nextInt();
        System.out.println("Enter name");
        name = in.next();
        System.out.println("Enter 3 subject marks");
        sub1 = in.nextInt();
        sub2 = in.nextInt();
        sub3 = in.nextInt();
    }

    void passOrFail() {
        if (sub1 < 50 || sub2 < 50 || sub3 < 50) {
            System.out.println("Fail");
            total = 0;
        } else {
            System.out.println("Pass");
            total = sub1 + sub2 + sub3;
        }
    }

    void display() {
        passOrFail();
        System.out.println("Enrollment Id: " + enrollId);
        System.out.println("Name: " + name);
        System.out.println("Sub1: " + sub1 + " Sub2: " + sub2 + " Sub3: " + sub3);
        System.out.println("Total: " + total);
    }
}
```

```
public class MainClass {  
    public static void main(String[] args) {  
        StudInfo s[] = new StudInfo[3];  
        for (int i = 0; i < s.length; i++) {  
            s[i] = new StudInfo();  
            s[i].display();  
        }  
    }  
}
```

Output:

Enter enrollment Id
101
Enter name
abc
Enter 3 subject marks
20 30 60

Fail
Enrollment Id: 101
Name: abc
Sub1: 20 Sub2: 30 Sub3: 60
Total: 0

Enter enrollment Id
102
Enter name
mno
Enter 3 subject marks
60 70 60

Pass
Enrollment Id: 102
Name: mno
Sub1: 60 Sub2: 70 Sub3: 60
Total: 190

Enter enrollment Id

103

Enter name

xyz

Enter 3 subject marks

50 50 50

Pass

Enrollment Id: 103

Name: xyz

Sub1: 50 Sub2: 50 Sub3: 50

Total: 150

7. In a college first year class have the following attributes Name of the class (BCA, BCom, BSc), Name of the staff, No of the students in the class, Array of students in the class. Define a class called first year with above attributes and define a suitable constructor. Also write a method called bestStudent() which processes a first-year object and returns the student with the highest total mark. In the main method define a first-year object and find the best student of this class

```
public class Student {
```

```
    int sID, totalMarks;
```

```
    Student(int id, int marks) {
```

```
        sID = id;
```

```
        totalMarks = marks;
```

```
    }
```

```
}
```

```
public class FirstYear {
```

```
    String cName, staffName;
```

```
    int noOfStuds;
```

```
    Student s[] = new Student[5];
```

```
    FirstYear(String cName, String staffName, int noOfStuds) {
```

```
        this.cName = cName;
```

```
        this.staffName = staffName;
```

```
        this.noOfStuds = noOfStuds;
```

```
        s[0] = new Student(101, 95);
```

```
        s[1] = new Student(102, 80);
```

```
        s[2] = new Student(103, 96);
```

```
        s[3] = new Student(104, 91);
```

```
        s[4] = new Student(105, 70);
```

```
    }
```

```
    Student bestStudent() {
```

```
        Student m = s[0];
```

```
        int max = s[0].totalMarks;
```

```
        for (int i = 1; i < 5; i++) {
```

```
            if (s[i].totalMarks > max) {
```

```
        max = s[i].totalMarks;  
        m = s[i];  
    }  
}  
return m;  
}  
}
```

```
package Student;  
  
public class MainClass {  
    public static void main(String args[])  
    {  
        FirstYear ob = new FirstYear("BCA", "ABC", 5);  
        Student p = ob.bestStudent();  
        System.out.println("Best Student");  
        System.out.println("ID: " + p.sID);  
        System.out.println("Total Marks: " + p.totalMarks);  
    }  
}
```

Output:

Best Student
ID: 103
Total Marks: 96

8. Write a Java program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.

```
import java.util.Date;
```

```
public class Employee {
```

```
    String name;
```

```
    Date dateOfAppointment;
```

```
    public Employee(String name, Date dateOfAppointment) {
```

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

```
        this.dateOfAppointment = dateOfAppointment;
```

```
    }
```

```
}
```

```
import java.util.Date;
```

```
public class EmployeeSort {
```

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

```
        Employee[] e = new Employee[10];
```

```
        e[0] = new Employee("abc", new Date(90, 0, 15));
```

```
        e[1] = new Employee("def", new Date(92, 2, 20));
```

```
        e[2] = new Employee("ghi", new Date(89, 11, 10));
```

```
        e[3] = new Employee("jkl", new Date(95, 5, 5));
```

```
        e[4] = new Employee("mno", new Date(88, 7, 25));
```

```
        e[5] = new Employee("pqr", new Date(90, 1, 18));
```

```
        e[6] = new Employee("stu", new Date(92, 7, 30));
```

```
        e[7] = new Employee("vwx", new Date(91, 10, 12));
```

```
        e[8] = new Employee("xyz", new Date(89, 8, 3));
```

```
        e[9] = new Employee("lmn", new Date(95, 3, 1));
```

```
        for (int i = 0; i < e.length - 1; i++) {
```

```
            for (int j = i + 1; j < e.length; j++) {
```

```
                if (e[i].dateOfAppointment.after(e[j].dateOfAppointment)) {
```

```

        Employee temp = e[i];
        e[i] = e[j];
        e[j] = temp;
    }
}

System.out.println("Employees sorted by date of appointment (seniority):");
for (int i = 0; i < e.length; i++) {
    System.out.println("-----");
    System.out.println(e[i].name);
    System.out.println(e[i].dateOfAppointment);
}
}
}

```

Output:

Employees sorted by date of appointment (seniority):

```

-----
mno
Thu Aug 25 00:00:00 IST 1988
-----
xyz
Sun Sep 03 00:00:00 IST 1989
-----
ghi
Sun Dec 10 00:00:00 IST 1989
-----
abc
Mon Jan 15 00:00:00 IST 1990
-----
pqr
Sun Feb 18 00:00:00 IST 1990
-----
vwx
Tue Nov 12 00:00:00 IST 1991
-----
def
Fri Mar 20 00:00:00 IST 1992
-----
stu

```

Sun Aug 30 00:00:00 IST 1992

lmn

Sat Apr 01 00:00:00 IST 1995

jkl

Mon Jun 05 00:00:00 IST 1995

9. Create a package 'student.Fulltime.BCA' in your current working directory
- Create a default class student in the above package with the following attributes: Name, age, sex.
 - Have methods for storing as well as displaying

```
package Student.FullTime.BCA;
```

```
public class Student {
```

```
    String name;  
    int age;  
    String sex;
```

```
    public Student(String n, int a, String s) {  
        name = n;  
        age = a;  
        sex = s;  
    }
```

```
    public void dispData() {  
        System.out.println("-----");  
        System.out.println("Name: " + name);  
        System.out.println("Age: " + age);  
        System.out.println("Sex: " + sex);  
    }  
}
```

```
package userdefinedpack;
```

```
import Student.FullTime.BCA.*;
```

```
public class UserDefinedPack {
```

```
    public static void main(String[] args) {  
        Student s1 = new Student("abc", 20, "Male");  
        Student s2 = new Student("mno", 21, "Female");  
        s1.dispData();  
        s2.dispData();  
    }
```

```
}
```


Output:

Name: abc

Age: 20

Sex: Male

Name: mno

Age: 21

Sex: Female

10. Write a program to demonstrate a division by zero exception

```
import java.util.Scanner;

public class DivByZeroExc {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter value a: ");
        int a = in.nextInt();
        System.out.println("Enter value b: ");
        int b = in.nextInt();
        try {
            int c = a / b;
            System.out.println("Quotient: " + c);
        }
        catch (ArithmeticException e) {
            System.out.println("Divided by zero operation not possible");
        }
    }
}
```

Output 1:

```
Enter value a:
10
Enter value b:
5
Quotient: 2
```

Output 2:

```
Enter value a:
10
Enter value b:
0
Divided by zero operation not possible
```

11. Write a program to create a user defined exception say PayOutOfBounds.

```
public class PayOutOfBoundsException extends Exception {  
    public PayOutOfBoundsException(String message) {  
        super(message);  
    }  
}
```

```
public class SalaryCalculator {  
    double MIN_PAY = 1000;  
    double MAX_PAY = 10000;  
  
    double calculateSalary(double pay) throws PayOutOfBoundsException {  
        if (pay < MIN_PAY || pay > MAX_PAY) {  
            throw new PayOutOfBoundsException("Pay is out of bounds. Pay must be between " + MIN_PAY + " and " + MAX_PAY);  
        }  
        return pay * 12;  
    }  
}
```

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

```
public class MainClass {  
  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        System.out.println("Enter amount to pay: ");  
        double pay = in.nextDouble();  
  
        SalaryCalculator s = new SalaryCalculator();  
        try {  
            double salary = s.calculateSalary(pay);  
            System.out.println("Annual Salary: " + salary);  
        } catch (PayOutOfBoundsException e) {  
            System.out.println("Error: " + e.getMessage());  
        }  
    }  
}
```

Output 1:

Enter amount to pay:

12000

Error: Pay is out of bounds. Pay must be between 1000.0 and 10000.0

Output 2:

Enter amount to pay:

8000

Annual Salary: 96000.0

12. Write a small program to catch Negative Array Size Exception. This exception is caused when the array size is initialized to negative values.

```
import java.util.Scanner;

public class NegArr {

    public static void main(String args[]) {
        int size;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the array size");
        try {
            size = in.nextInt();
            if (size < 0) {
                throw new NegativeArraySizeException("Array size cannot be negative");
            } else {
                int a[] = new int[size];
                System.out.println("Array Created");
            }
        } catch (NegativeArraySizeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Output 1:

```
Enter the array size
5
Array Created
```

Output 2:

```
Enter the array size
-5
Array size cannot be negative
```

13. Write a program to handle Null Pointer Exception and use the "finally" method to display a message to the user.

```
public class MainClass {  
    public static void main(String[] args) {  
        String str = null;  
  
        try {  
            int length = str.length();  
            System.out.println("Length of the string: " + length);  
        } catch (NullPointerException e) {  
            System.out.println("NullPointerException occurred: " + e.getMessage());  
        } finally {  
            System.out.println("Finally block executed. End of program.");  
        }  
    }  
}
```

Output:

NullPointerException occurred: Cannot invoke "String.length()" because "str" is null
Finally block executed. End of program.

14. Write a program which creates and displays a message on the window. Write a program to draw several shapes in the created window

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

```
public class DrawingPanel extends JPanel {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.BLUE);
        g.drawRect(50, 100, 100, 80);
        g.drawOval(200, 100, 100, 80);
        g.drawLine(50, 250, 300, 250);
    }
}
```

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

```
public class DrawShapesInWindow {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Drawing Shapes");
        JLabel label = new JLabel("Hello, Window!");
        frame.getContentPane().add(label, BorderLayout.NORTH);
        DrawingPanel drawingPanel = new DrawingPanel();
        frame.getContentPane().add(drawingPanel, BorderLayout.CENTER);
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

