

Date :-

Exp No:-  
① a)

Q) Write a Java Program to demonstrate Constructor Overloading

Program:-

class Constructor Overloading

{ int rollno;

String name;

Constructor Overloading()

{

System.out.println("No argument constructor");

}

Constructor Overloading(int rollno)

{

System.out.println("1 Argument constructor");

}

Constructor Overloading(int rollno, String name)

{

System.out.println("2 argument constructor");

}

Constructor Overloading(String name, int rollno)

{

System.out.println("Different type argument constructor");

}

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

```
{
```

```
    Constructor Overloading c1 = new Constructor Overloading ();
```

```
    Constructor Overloading c2 = new Constructor Overloading (3360);
```

```
    Constructor Overloading c3 = new Constructor Overloading (3360, "Yash");
```

```
    Constructor Overloading c4 = new Constructor Overloading ("Yash", 3360);
```

```
}
```

```
}
```

Output:-

No argument constructor

1 argument constructor

2 argument constructor

Different type argument constructor.

(1) friend class

((reference temporary in " ) returning to

(using this) friend class

((reference temporary in " ) returning to

(using this) friend class

((reference temporary in " ) returning to

(using this) friend class

((reference temporary in " ) friend class

((reference temporary in " ) friend class

CLERK'S OFFICE OF THE STATE OF PENNSYLVANIA  
RECEIVED IN THE CLERK'S OFFICE OF THE STATE OF PENNSYLVANIA

Output:-

50

50

2

45.632

45

45

(gpa [ ] per m<sup>2</sup>) min bio

abt 2000 stonewall area = 12 pikelets

abt 2000 stonewall area = 13 pikelets

abt 2000 stonewall area = 12 pikelets

abt 2000 stonewall area = 12 pikelets

Date:-

Exp no:-  
(1 c)Program:-

class Bitwise

{ public static void main (String [] args )

{ int a = 5 , b = 7;

System.out.println ("a+b = " + (a+b));

System.out.println ("a|b = " + (a|b));

System.out.println ("a ^ b = " + (a ^ b));

System.out.println ("~a = " + (~a));

System.out.println ("a&lt;&lt;2 = " + (a&lt;&lt;2));

System.out.println ("a&gt;&gt;2 = " + (a&gt;&gt;2));

System.out.println ("a&gt;&gt;&gt;2 = " + (a&gt;&gt;&gt;2));

}

}

Output:-

$$a \& b = 5$$

$$a | b = 7$$

$$a ^ b = 2$$

$$\sim a = -6$$

$$a << 2 = 20$$

$$a >> 2 = 1$$

$$a >>> 2 = 1$$

signed on the portion left of the sign (left)  
works on the portion right of the sign (right)

the sign of what is prime sign of the number  
Hyp of what is prime sign of the number

- (a) wrong. true
- (b) wrong. true
- (c) wrong. true
- (d) wrong. true
- (e) wrong. true
- (f) wrong. true
- (g) wrong. true

Date:-

Exp No:-

(1) a)

Q) Write a program to demonstrate "this" keyword.

Program:-

class Student

{ int rollno;

String name;

Student (int rollno, String name)

{ this.rollno = rollno;

this.name = name;

}

public static void main (String [] args)

{ Student s = new Student (3360, "Yash");

System.out.println (s.rollno);

System.out.println (s.name);

}

y

Output:-

3360

Yash

(open [3 part]) main body ok

((d10) + " - d10") writing - two

Date :-

Exp No:-

(2)

Q) Write a java program to demonstrate static polymorphism (Method Overloading).

Program:

```

class Method Overloading
{
    static void sum( )
    {
        System.out.println ("Method with zero parameters");
    }

    static int sum (int x, int y)
    {
        return x+y;
    }

    static double sum (int x, double d)
    {
        return x+d;
    }

    static double sum (double d, int u)
    {
        return n+d;
    }

    static int sum (int x, int y, int z)
    {
        return x+y+z;
    }
}

```

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

{

```
    Method Overloading . sum ();
```

```
    System.out.println (Method Overloading . sum (2,4));
```

```
    System.out.println (Method Overloading . sum (2,2,3));
```

```
    System.out.println (Method Overloading . sum (2,3,4));
```

```
    System.out.println (Method Overloading . sum (2,4,6));
```

}

y

Output:-

6

4.3

6.3

12

shells afterwards at majority every so often  
(prioritizing bottom) majority

grabs other

(bottom) majority  
and when bottom of majority  
is taken

(prioritizing) own

(bottom) majority  
(bottom) own shell

(own shell) own shell  
bottom shell

(bottom) own shell  
bottom shell

Date:-

Exp No:-

(3)

Q) Write a program to understand full capability of String class. Implement as many methods as required.

Page No. 8

### class String Methods

```
{ public static void main (String args[])
{
    String s = "Hello World";
    System.out.println (s.contains ("Hell"));
    System.out.println (s.replace ('l', 'm'));
    System.out.println (s.indexOf ('w'));
    String s2 = "Java Programming";
    System.out.println (s.compareTo (s2));
    String s3 = "Hello world";
    System.out.println (s.compareToIgnoreCase (s3));
    System.out.println (s.trim ());
    System.out.println (s2.toLowerCase ());
    System.out.println (s2.replace ("Programming", "Java"));
    String s5 = "";
    System.out.println (s5.isEmpty ());
}
```

Output:-

true

Hemmo World

6

-2

0

Hello World

java programming

Java Java

false

e:-

o)

NO:-

2

③

is String

ublic sta

String s

System.out

System.out

System.out

String s2

System.out

String s3

System.out

System.out

System.out

System.out

String s

System.out

Date :-

Expt No:-

(4)

Page No. 9

Develop a java application to demonstrate  
String Buffer constructors and methods.

import java.io.\*;

class StringBuffer Demo

{ public static void main (String [ ] args)

{

StringBuffer s = new StringBuffer ("mess for buss");

int p = s.length();

int q = s.capacity();

System.out.println ("Length of String mess for buss = " + p);

System.out.println ("capacity of String mess for buss = " + q);

}

3

CVR COLLEGE OF ENGINEERING

Vastunagar, Mangalpalli (V) Ibrahimpatan (M), R.R. Dist. Ph - 501 510

Output:-

length of string mess for ltrs = 13

capacity of string mess for ltrs = 29

Date :-

Expt:-  
⑤

Q. Write a Java program to illustrate the inheritance concept (Ex. Hierarchy of employees in a University).

Program:-

class Principal {

    void display1() {

        System.out.println("In Principal class");

}

class HOD extends Principal {

    void display2() {

        System.out.println("In HOD class");

}

class Professor extends HOD {

    void display3() {

        System.out.println("In Professor class");

}

class EmployeeDemo {

    public static void main (String[] args)

{

    Professor P = new Professor();

    P.display1();

    P.display2();

    P.display3();

}

}

Output:-

- In Principal class
- In HOD class
- In Professor class

le :-

i:-

5)

> gen:-

ss Principal  
void d

S

3

> HOD e  
void d

Su

3

> Professor  
void d

S

> Employee  
public  
{  
Proff

Proff

3

Date:-

Expt No:-  
⑥

Page No. : 11

Q) Write a Java program to demonstrate the usage of  
a) Super      b) final (affid of access modifier a  
inheritance).

Class Principal

{ void display() {

    System.out.println("In Principal class");

}

}  
final class HOD extends Principal {

    void display() {

        System.out.println("In H.O.D class");

}

}  
class Professor extends HOD {

    void display() {

        System.out.println("In Professor class");

}

}  
class SuperDemo {

    public static void main (String [] args)

{

    Professor p = new Professor();

    p.display();

    p.display();

    p.display();

}

}

Output:-

error 1 as Final class HOD cannot extend Principal

fei:-	②) W
p No:-	of ⑥ in

as Principal  
void display  
System

3

3 beginning address  
3 as per  
(and) OOH at \*) memory location

al class HOD  
void disp  
Sys

3

3 OOH address  
3 as per

to Professor  
void d  
S  
3

as SuperDemo  
public stat  
2

Professor  
p. di  
p. di  
p. di

y

Date :-

Exp No.:  
7

Q) Write a Java program to demonstrate dynamic polymorphism (method overriding with dynamic method dispatch).

Class Employee {

    void display () {

        System.out.println ("In Employee Method");

{}

Class Manager extends Employee {

    void display () {

        System.out.println ("In Manager Method");

{}

Class DynamicMethod Dispatch Demo {

    public static void main (String args)

{

        Employee e = new Employee ();  
            e.display ();

        Manager m = new Manager ();  
            m.display ();

        Employee e1 = new Manager ();  
            e1.display ();

{}

Output:-

In Employee Method  
In Manager Method  
In Manager Method

De :-

No:-  
⑦

ass Employee  
void dis

Sys  
y

ass Manager  
void dis

Sys  
y

us Dynamic  
public

{  
Employee

Manager

Employee

(no 5 part 2) main fun

if (c) part  
(c) part  
(c) part

Vastuna

Date:-

Exp No:-  
③ a)

Q) Write a java program to demonstrate abstract classes

abstract class Person {

    abstract void display();

    void show() {

        System.out.println("In Person class");

}

class Faculty extends Person {

    void display() {

        System.out.println("In Faculty class");

}

}

    void display() {

        System.out.println("In Student class");

}

}

    public static void main (String args) {

        Faculty f = new Faculty();

        f.display();

        f.show();

        Student s = new Student();

        s.display();

        s.show();

}

}

Output:-

In Faculty class  
In Person class  
In Student class  
In Person class

all:-  
in No:-  
② a)

student do  
abstract  
void

y

ns Faculty  
void  
System

y

ns Student  
void

y

y

ns Abstract  
public

y

y

Date:-

Expt No:-  
⑧ b)Q) Write a Java Program to demonstrate the  
use of Interfaces.

interface Polygon

{ void getArea (int length, int breadth); }

class Rectangle implements Polygon

{ public void getArea (int length, int breadth) {

System.out.println ("The area of the Rectangle is " +  
(length \* breadth)); }

class InterfaceDemo

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

Rectangle r1 = new Rectangle ();  
r1.getArea (5, 6); }

}

}

Output:-

The Area of the Rectangle is 30.

Date:-

Q No:-  
8) b)

Surface Po  
- Void

class Rect

Public  
2

Sys+

as Interf

public sta  
2 Recta

3

Date:-

Expt No:-

⑨

Q) Demonstrate how to create and access passages in Java (Custom Passages / Standard passages).

# Andhra Pradesh

passage com.cvr.indianities;  
public class AndhraPradesh

{ public void stateCapital()

{ System.out.println ("The capital of Andhra Pradesh is  
Amaravati");

}

{ # Telengana

passage com.cvr.indiancities;  
public class Telengana {

public void stateCapital() {

System.out.println ("The capital of Telengana is Hyderabad");

}

{ # Maharashtra

passage com.cvr.indianities;  
public class TelenganaMaharashtra {

public void stateCapital() {

System.out.println ("The capital of Maharashtra is  
Mumbai");

}

main Doc  
import com.cvr.indiancities.\*;  
class States

{ public static void main (String args[])

{ AndhraPradesh a = new AndhraPradesh ();  
a.stateCapital();

Telangana t = new Telangana ();  
t.stateCapital();

Maharashtra m = new Maharashtra ();  
m.stateCapital();

}

Output:-

The Capital of Andhra Pradesh is Amaravati  
The Capital of Telangana is Hyderabad  
The Capital of Maharashtra is Mumbai

# main doc  
import com  
class Stat  
2 public  
2

Andhra

Telangana

Maharashtra

y

y

with its capital - 200 km  
from the old capital  
of Hyderabad state.  
Capital of Andhra  
and Telangana states  
is located at  
Hyderabad city.

Q) Develop a java program illustrating try-catch and finally.

Ans) Exception Demo :

```

public static void main (String[] args) {
    int [] arr = new int [4];
    try {
        int i = arr [4];
        System.out.println ("Inside try block");
    }
    catch (ArrayIndexOutOfBoundsException ex) {
        System.out.println ("Exception caught in catch
                            block");
    }
    finally {
        (finally block executed)
        System.out.println ("Finally block executed");
    }
    System.out.println ("Outside try - catch
                        finally block");
}

```

Output:-

Exception caught in catch block

finally block executed

Outside try - catch - finally clause

Date:-

Expt No:  
10) a)

class

3

Date:-  
Page No.:-  
10 b)

Q) Develop a java program illustrating throws.

class ThrowsExcp {

    static void fun() throws IllegalAccessException {

        System.out.println("Inside fun().");

        throw new IllegalAccessException("demo");

    }

}

    try {

        fun();

    }

    catch (IllegalAccessException e) {

        System.out.println("Caught in main.");

    }

}

CVR COLLEGE OF ENGINEERING

Vastunagar, Mangalpalli (V) Ibrahimpatan (M), R.R. Dist. Ph - 501 510

Output:

Inside func()

Caught in main.

[1] 3 \* 500 = 1500

(and put whatever) always two messages

(for example, "Hello World" prints twice)

(and whatever output) always two messages

"Hello"

following

(Hello with printf) because each printf

does printf) always two messages

3

Q) Develop a java program illustrating nested try.

Ans Nested Try ↴

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

try {

```
    int a[] = {1, 2, 3, 4, 5};
```

try {

```
    int x = a[2] / 0;
```

}

Catch (Arithmetic Exception e) {

```
    System.out.println ("Division by zero is  
not possible");
```

}

}

Catch (Array Index Out of Bounds Exception e)

{

```
    System.out.println ("Array Index Out of Bounds Exception");
```

```
    System.out.println ("Element at such index does not  
exists");
```

}

}

Output:-

Array Index Out of Bounds Exception  
Element at such index does not exist.

Develop a Java program illustrating multi catch.

Page No. 20

```
public class MultipleCatchBlock {  
    public static void main(String[] args) {  
        try {  
            int a[] = new int[5];  
            a[5] = 30/0;  
        }  
        catch (ArithmeticException e)  
        {  
            System.out.println("Arithmetic Exception occurs");  
        }  
        catch (ArrayIndexOutOfBoundsException e)  
        {  
            System.out.println("ArrayIndexOutOfBoundsException  
            occurs");  
        }  
        catch (Exception e)  
        {  
            System.out.println("Parent Exception occurs");  
        }  
        System.out.println("rest of the code");  
    }  
}
```

Output:-

Aritmetic Euphia Daws  
nest of the lode.

( $\text{C} + \text{open point}$ ) min. time solution  $\rightarrow$  3 p.m.

$\text{C} = 2000 \times 3 = 30000 \rightarrow$  3 p.m.

$\therefore [s]D = x \text{ km}$

$\therefore$  (2 hours westward, Arithmetical note)

{(forward) after warning?  
slowing down

(or vice versa) without passing) note

backward for 3 hours per sec " (stop & wait)  
forward with same no time") after breaking  
(slow)

Q) Develop a java program illustrating creation  
of user-defined packages in java.

import java.util.Arrays;

JavaUtil Example {

public static void main(String args[])

{

int[] intArray = {10, 30, 20, 50, 40};

Arrays.sort(intArray);

System.out.println("Sorted array : - " + Arrays.toString(intArray));

}

**CVR COLLEGE OF ENGINEERING**

Vastunagar, Mangalpalli (V) Ibrahimpatan (M), R.R. Dist. Ph - 501 510

Output:

Sorted array [10, 20, 30, 40, 50]

Time complexity: O(n log n) (n = number of elements)

Space complexity: O(1) (constant space)

Stable sort: Yes

Comparison based: Yes

Comparison count:  $\frac{n(n-1)}{2}$  (number of comparisons)

Number of steps:  $n \log n$  (number of steps)

Number of nodes:  $n$  (number of nodes)

Number of levels:  $\log n$  (number of levels)

Number of branches:  $n$  (number of branches)

Number of leaves:  $n$  (number of leaves)