**VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES**

**VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated with medium confidence

**MASTERS OF COMPUTER APPLICATION**

**JAVA PRACTICAL FILE**

**MCA-167**

**Guru Gobind Singh Indraprastha University Sector - 16C Dwarka, Delhi – 110078**

Logo

Description automatically generated

**SUBMITTED TO: SUBMITTED BY:**

Dr. Neha Verma Pranjal Tiwari

Assistant Professor 12017704424

VSIT MCA – 1B

Q1 Write a Java program to print all odd numbers between 1 to 10.

Code.

package javafile;

public class ques\_1 {

public static void main(String[] args) {

for (int i = 1; i < 10; i++) {

if (i % 2 != 0) {

System.out.println(i);

}

}

}

}

Output

A computer screen shot of a computer code

Description automatically generated

Q2 Write a Java program to find out factorial of a number through recursion

Code.

package javafile;

import java.util.\*;

public class ques\_2 {

public static int fact(int n) {

if(n==1){

return 1;

}

else{

return n\*fact(n-1);

}

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.print("enter a number = ");

int n=sc.nextInt();

int ans=fact(n);

System.out.println("factorial of "+n+" is = "+ans);

}

}

Output

A screen shot of a computer

Description automatically generated

Q3 Write a Java program to accept command line arguments &amp; print them

Code.

package javafile;

public class ques\_3 {

public static void main(String args[]){

for(int i=0;i<args.length;i++)

System.out.println(args[i]);

}

}

Output

A computer screen with text

Description automatically generated

Q4 Write a Java program to print fibonacci series.

Code.

package javafile;

import java.util.\*;

public class ques\_4 {

public static int fib(int n) {

if(n==1 ){

return 0;

}

if(n==2){

return 1;

}

else{

return fib(n-1) + fib(n-2);

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("enter a number = ");

int num=sc.nextInt();

for(int i=1;i<=num+1;i++){

System.out.print(fib(i)+" ");

}

}

}

Output

A computer screen shot of a computer code

Description automatically generated

Q5 WAP that creates a class Accounts with following details:Instance variables: ac\_no., name, ac\_name, balance .Methods: withdrawal(), deposit(),display().Use constructors to initialize members.

Code.

class Accounts {

long account\_number;

String name;

String account\_name;

double balance;

Accounts(long account\_number, String name, String account\_name, double balance) {

this.account\_number = account\_number;

this.name = name;

this.account\_name = account\_name;

this.balance = balance;

}

void withdrawal(int amount) {

balance -= amount;

System.out.println("Amount successfully withdrawled, now the current balance is = "+ balance);

}

void deposit(int amount) {

balance += amount;

System.out.println("Amount deposited successfully , now the current balance is = "+balance);

}

void display() {

System.out.println("Account no = " + account\_number);

System.out.println("Account holder name = " + name);

System.out.println("Account type = " + account\_name);

System.out.println("Balance = " + balance);

}

}

public class ques\_5 {

public static void main(String[] args) {

Accounts obj = new Accounts(120, "Pranjal Tiwari", "Saving Account", 46000);

obj.withdrawal(7600);

obj.deposit(8800);

obj.display();

}

}

Output

A computer screen with blue and red text

Description automatically generated

Q6 Write a Java program to implement constructor overloading

Code.

class overload{

overload(){

System.out.println("performing function overloading");

}

overload(int a){

System.out.println("enrollment no. is = "+a);

}

}

class ques\_6 {

public static void main(String[] args) {

overload obj=new overload();

overload obj1 =new overload(120);

}

}

Output

A computer screen with colorful text

Description automatically generated

Q7 Write a Java program to count the no. of objects created in a program

Code.

class counter{

static int count;

counter(){

count++;

System.out.println("object number = "+count);

}

}

public class ques7 {

public static void main(String[] args) {

counter obj1=new counter();

counter obj2=new counter();

counter obj3=new counter();

counter obj4=new counter();

counter obj5=new counter();

}

}

Output

Text

Description automatically generated

Q8 WAP to show call by value & call by reference.

Code.

package javafile;

public class ques\_8 {

public static void call\_by\_value(int number){

number = number+1; // increment variable by 1

System.out.println("value in method: "+number);

}

int number=15;

public static void call\_by\_reference(ques\_8 obj){

obj.number = obj.number+1; // increment variable by 1

System.out.println("value in method: "+obj.number);

}

public static void main(String[] args) {

int number=10;

System.out.println("value before method call : "+number);

call\_by\_value(number);

System.out.println("value after method call: "+number);

ques\_8 obj=new ques\_8();

System.out.println("value before method call: "+obj.number);

call\_by\_reference(obj);

System.out.println("value after method call: "+obj.number);

}

}

Output

A computer screen with white text

Description automatically generated

Q9. WAP to implement method over ridding & method overloading.

Code.

class overload{

public void show(){

System.out.println("implementing overloading function in java");

}

public void show(String Name){

System.out.println("student name = "+Name);

}

}

class override extends overload{

public void show(){

System.out.println("implementing overriding function in java" );

}

}

public class ques\_9 {

public static void main(String[] args) {

overload obj=new overload();

obj.show();

obj.show("Pranjal");

override obj1=new override();

obj1.show();

}

}

Output

A screen shot of a computer

Description automatically generated

Q 10. Create a class box having height, width, depth as the instance variables &amp;calculate its volume. Implement constructor overloading in it. Create a subclass named box\_new that has weight as an instance variable. Use super in the box\_new class to initialize members of the base class

Code.

class Box {

int height, width, depth;

Box(){

System.out.println("using default Constructor ");

}

Box(int height, int width, int depth) {

this.height = height;

this.width = width;

this.depth = depth;

}

void volume() {

System.out.println("The volume of box is " + (height \* width \* depth));

}

}

class Box\_new extends Box{

int weight;

Box\_new(){

super(16,10,12);

weight = 10;

}

}

public class ques\_10 {

public static void main(String[] args) {

Box obj1 = new Box();

Box obj2 = new Box(13,9,11);

Box\_new obj3 = new Box\_new();

obj2.volume();

obj3.volume();

}

}

Output

A screen shot of a computer

Description automatically generated

Q 11 Write a Java program to implement run time polymorphism

Code.

class Vehicle {

public void drive() {

System.out.println("Driving a vehicle");

}

}

class Car extends Vehicle {

public void drive() {

System.out.println("Driving a car");

}

}

class Bus extends Vehicle {

public void drive() {

System.out.println("Driving a bus");

}

}

class Truck extends Vehicle {

public void drive() {

System.out.println("Driving a truck");

}

}

public class ques\_11 {

public static void main(String[] args) {

Vehicle obj = new Vehicle();

Vehicle obj1 = new Car();

Vehicle obj2 = new Bus();

Vehicle obj3 = new Truck();

obj.drive();

obj1.drive();

obj2.drive();

obj3.drive();

}

}

Output

A screen shot of a computer

Description automatically generated

Q 12. Write a Java program to implement interface. Create an interface named shape having area () &amp; perimeter () as its methods. Create three classes circle, rectangle &amp; square that implement this interface

Code.

interface shape {

void area();

void perimeter();

}

class circle implements shape{

double pi;

int rad;

circle(double pi, int rad){

this.pi = pi;

this.rad = rad;

}

public void print(){

System.out.println("area and perimeter of circle");

}

public void area() {

System.out.println( "area = "+pi\*rad);

}

public void perimeter(){

System.out.println("perimeter = "+2\*pi\*rad);

}

}

class rectangle implements shape{

int len,wid;

rectangle(int len, int wid){

this.len = len;

this.wid = wid;

}

public void print(){

System.out.println("area and perimeter of rectangle");

}

public void area() {

System.out.println("area = "+len\*wid);

}

public void perimeter(){

System.out.println("perimeter = "+2\*len\*wid);

}

}

class square implements shape{

int side;

square(int side){

this.side = side;

}

public void print(){

System.out.println("area and perimeter of square");

}

public void area() {

System.out.println("area = "+side\*side);

}

public void perimeter(){

System.out.println("perimeter = "+2\*side);

}

}

public class ques\_12 {

public static void main(String[] args) {

circle obj1 = new circle(3.14, 9);

obj1.print();

obj1.area();

obj1.perimeter();

rectangle obj2 = new rectangle(7, 4);

obj2.print();

obj2.area();

obj2.perimeter();

square obj3 = new square(5);

obj3.print();

obj3.area();

obj3.perimeter();

}

}

Output

A screenshot of a computer program

Description automatically generated

Q 13. Write a Java program to show multiple inheritance.

Code.

interface A

{

public void func\_A();

}

interface B

{

public void func\_B();

}

class C implements A,B

{

public void func\_A()

{

System.out.println("calling function func\_A");

}

public void func\_B()

{

System.out.println("calling function func\_B");

}

}

public class ques\_13

{

public static void main(String[] args)

{

C obj = new C();

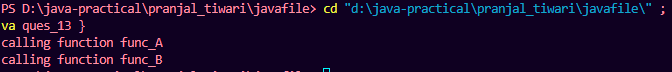
obj.func\_A();

obj.func\_B();

}

}

Output



Q 14. Write a Java program to implement exception handling. Use try, catch & finally

Code.

package javafile;

public class ques\_14 {

public static void main(String[] args) {

int num = 76;

int den = 0;

try {

int ans = num / den;

System.out.println("ans is " + ans);

}

catch (ArithmeticException e) {

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

}

finally {

System.out.println("Execution successfull.");

}

}

}

Output

A black background with red and white text

Description automatically generated

Q 15 . Write a Java program to implement matrix multiplication by 2d array

Code.

Output

Q 16. Write a Java program to implement vector [use: addelement(), elementat(),removeElement(),size().]

Code

package javafile;

import java.util.*\**;

public class ques\_16 {

    public static void main(String[] args) {

        Vector<String> vector\_arr = new Vector<>();

        vector\_arr.addElement("JAVA");

        vector\_arr.addElement("DBMS");

        vector\_arr.addElement("DS");

        vector\_arr.addElement("OS");

        vector\_arr.addElement("CN");

        System.out.println("vector\_arr after adding elements = " + vector\_arr);

        System.out.println("Element at index 2 = " + vector\_arr.elementAt(2));

        System.out.println("Element removed = "+  vector\_arr.removeElement("CN"));

        System.out.println("Size of the vector\_arr: " + vector\_arr.size());

        System.out.println("vector\_arr =   " + vector\_arr);

    }

}

Output

A computer screen with text

Description automatically generated

Q17. Create a user defined exception named “nomatchexception” that is fired when the string entered by the user is not “india”.

Code.

import java.util.\*;

class NoMatchException extends Exception {

public NoMatchException(String message) {

super(message);

}

}

public class ques\_17 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a string = ");

String input = sc.next();

try {

if (!input.equals("india")) {

throw new NoMatchException("Input does not match 'india'.");

}

System.out.println("Input matches 'india'.");

}

catch (NoMatchException e) {

System.out.println(e.getMessage());

}

finally {

System.out.println("execution completed");

}

}

}

Output

A computer screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated