Assignment-2

1. Write a java program for Matrix Addition.

Ans:-

```
Import java.util.*;
 public class A{
 public static void main(String args[]){
 Scanner sc=new Scanner(System.in);
 int n=sc.nextInt();
 int A[][]=new int [n][n];
  int B[][]=new int [n][n];
 int C[][]=new int [n][n];
 for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
  A[i][j]=sc.nextInt();
  }
  }
   for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
  B[i][j]=sc.nextInt();
  }
  }
  for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
  C[i][j]=A[i][j]+B[i][j];
  }
  }
  for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
    System.out.print(C[i][j]+ " ");
  }
  System.out.println(" ");
  }
 }
 }
  Output:-
 3 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
 11 13 15
 17 19 21
23 25 27
```

2. Write a java program for Matrix Multiplication

Program:-

```
Import java.util.*;
public class Mul{
 public static void main(String args[]){
Scanner sc=new Scanner(System.in);
Int n=sc.nextInt();
 int A[][]=new int [n][n];
  int B[][]=new int [n][n];
 int C[][]=new int [n][n];
 for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
  A[i][j]=sc.nextInt();
  }
  }
   for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
  B[i][j]=sc.nextInt();
  }
  }
  for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
 // c[i][j]=0;
```

```
   for(int k=0;k<n;k++){
  C[i][j]+=A[i][k]+B[k][j];
  }
  }
  }
  for(int i=0;i<n;i++){
  for(int j=0;j<n;j++){
          System.out.print(C[i][j]+ " ");
  }
  System.out.println(" ");
  }
 }
 }
```

Output:-

```
 3 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
45 48 51
54 57 60
63 66 69
```

3. Write a java program to demonstrate method overloading.

Program:-

```
class MethodOverloading{
static int add(int a,int b){
return a+b;
}
static int add(int a,int b,int c){
return a+b+c;
}
public static void main(String[] args){
MethodOverloading M=new MethodOverloading();
System.out.println(M.add(1,2));
System.out.println(M.add(1,2,3));
}}
```

Output:-

```
3
6
```

4. Write a java program to create a class Point with two data members x & y. Include all constructors and display().

Ans:-

Program:-

```
import java.lang.*;
class Point
{
 int x,y;
public Point()
{
 x=10;
 y=20;
}
public Point(int r)
{
 x=r;
 y=20;
}

public int main()
{
  return x+y;
}

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

```
Point t1=new Point();
System.out.println(“default constructor or no argument constructor");
System.out.println(t1.main());
Trail t2=new Trail(15);
System.out.println("parameterized constructor");
System.out.println(t1.main());
```

Output:-

```
Default constructor or no argument constructor
30
Parameterized constructor
45
```

5. Write a java program using static method.

Program:-

```
class Static{
  static int main(int x){
  return x;
  }
 public static void main(String args[]){
  int result=main(10);
  System.out.println(result);
  }  }
```

Output:-

```
10
```

1.What is conditional statement?

Ans:- A conditional statement is a statement that computer programming language used to decide which code has to be run when the true condition is met or which code has not to be run when the true condition is not met.

Let's see the following conditional statements

1. if statement
2. nested if statement
3. if-else statement
4. if-else-if statement
5. Switch Case Statement

1.if Statement:-The if statement is the most basic of all the control flow statements. The if statement tells our program to execute a certain section of code only if a particular test evaluates to true.

Syntax:-

```
if(condition){
  Statement(s);
}
```

2.Nested if statement: An if statement inside another the statement. If the outer if condition is true then the section of code under outer if condition would execute and it goes to the inner if condition. If inner if condition is true then the section of code under inner if condition would execute.

Syntax:-

```
if(condition_1) {
   Statement1(s);

   if(condition_2) {
      Statement2(s);
   }
}
```

3.if-else Statement:-

If a condition is true then the section of code under if would execute else the section of code under else would execute.

Syntax:-

```
if(condition_1) {
  /*if condition_1 is true execute this*/
  statement(s);
}
else if(condition_2) {
  /* execute this if condition_1 is not met and
   * condition_2 is met
   */
  statement(s);
}
else if(condition_3) {
  /* execute this if condition_1 & condition_2 are
   * not met and condition_3 is met
   */
  statement(s);
}
else {
  /* if none of the condition is true
   * then these statements gets executed
   */
  statement(s);

}
```

4.if-else –if statement:-

The if-else-if ladder statement executes one condition from multiple statements.

Syntax:-

```
if(condition_1) {
  /*if condition_1 is true execute this*/
  statement(s);
}
else if(condition_2) {
  /* execute this if condition_1 is not met and
   * condition_2 is met
   */
  statement(s);
}
else if(condition_3) {
  /* execute this if condition_1 & condition_2 are
   * not met and condition_3 is met
   */
  statement(s);
}
.
.
.
else {
  /* if none of the condition is true
   * then these statements gets executed
   */
  statement(s);
}
```

5.Switch Case:-

The switch statement in Java is a multi branch statement. We use this in Java when we have multiple options to select. It executes particular option based on the value of an expression.

Syntax:-

```
switch(expression) {
   case valueOne:
      //statement(s
      break;
   case valueTwo:
      //statement(s
      break;
   default: //optional
```

```
    //statement(s) //This code will be executed if all cases are not matched}
```

2.Write the syntax of switch..Case statement.

Ans:-
```
switch(expression) {
    case valueOne:
        //statement(s
        break;
    case valueTwo:
        //statement(s
        break;
          :
          :
    default: //optional
    //statement(s) //This code will be executed if all cases are not matched
}
```

3. Write the difference between break and continue statement.

Ans:-

‘break' stops the continuation of loop.It terminates the execution of remaining iteration of the loop

'continue' do not stop the continuation of loop, it only stops the current iteration.

4. What is looping statement?

Ans:-Looping statement are the statements execute one or more statement repeatedly several number of times. In java programming language there are three types of loops; while, for and do-while.

ADVANTAGES OF LOOP

- Reduce length of Code
- Take less memory space.
- Burden on the developer is reducing.
- Time consuming process to execute the program is reduced.

5. Write the difference between while and do..while statement.

In while loop statements may not be executed even once.
booleanVal is evaluated at top in while loop.So, Statements inside while loop are executed when booleanVal is true.
In do-while loop statements are at least executed once.
First statements inside do block are executed, then booleanVal is evaluated.

6. What is array? How it is created?

Ans:- Java provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables.
Declaring Array Variables:-
To use an array in a program, you must declare a variable to reference the array, and you must specify the type of array the variable can reference. Here is the syntax for declaring an array variable
Syntax:-

```
dataType[] arrayRefVar;   // preferred way.
or
dataType arrayRefVar[];  // works but not preferred way.
```

7. What is class?

Ans:- A class is a blueprint from which individual objects are created.

```
class <class_name>{
   field;
   method;
 }
```

8. What is constructor?

Ans:- A constructor initializes an object when it is created. It has the same name as its class and is syntactically similar to a method. However, constructors have no explicit return type.

Typically, you will use a constructor to give initial values to the instance variables defined by the class, or to perform any other start-up procedures required to create a fully formed object.

All classes have constructors, whether you define one or not, because Java automatically provides a default constructor that initializes all member variables to zero. However, once you define your own constructor, the default constructor is no longer used.

Syntax:-

```
Class Classname{
    Classname(){
}
}
```

Java allows two types of constructors namely −

- No argument Constructors
- Parameterized Constructors

No argument Constructors

As the name specifies the no argument constructors of Java does not accept any parameters instead, using these constructors the instance variables of a method will be initialized with fixed values for all objects.

Parameterized Constructors

Most often, you will need a constructor that accepts one or more parameters. Parameters are added to a constructor in the same way that they are added to a method, just declare them inside the parentheses after the constructor's name.

9. What is the use of a copy constructor?

Ans:- A copy constructor in a Java class is a constructor that creates an object using another object of the same Java class. That's helpful when we want to copy a complex object that has several fields, or when we want to make a deep copy of an existing object.

10. What is the use of this keyword?

Ans: - this keyword in Java is mainly used to refer to the current instance variable of the class. It can also be used to implicitly invoke the method or to invoke the constructor of the current class.

11. What is method overloading?

Ans: - When a class has two or more methods by the same name but different parameters, at the time of calling based on the parameters passed respective method this mechanism is known as method overloading.

12. What is a static variable?

Ans: - The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

Static variable:-

If you declare any variable as static, it is known as a static variable.

· The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.

· The static variable gets memory only once in the class area at the time of class loading.

13. What is an access modifier?

Ans:- There are two types of modifiers in Java: access modifiers and non-access modifiers.

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

1. Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. Default: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.
3. Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
4. Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

14. Write the difference between instance and static methods.

Ans:-

A static method belongs to the class, and you do not have to create an instance of the class to access the static method.

A non-static method belongs to an object of the class, and you have to create an instance of the class to access the non-static method.

15. What is an object ? How is it created?

Ans:- Object − Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors – wagging the tail, barking, eating. An object is an instance of a class

Creating an Object

A class provides the blueprints for objects. So basically, an object is created from a class. In Java, the new keyword is used to create new objects.

There are three steps when creating an object from a class −

- Declaration − A variable declaration with a variable name with an object type.
- Instantiation − The 'new' keyword is used to create the object.
- Initialization − The 'new' keyword is followed by a call to a constructor. This call initializes the new object.