

ASSIGNMENT-2

1. Write a java program for Matrix Addition

```
public class Matrixaddition {  
    public static void main (String args []){  
        int n=Integer.parseInt(args [0]);  
        int c=1;  
        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]=Integer.parseInt(args[c]);  
                c=c+1;  
            }  
        }  
        for (int i=0; i<n; i++) {  
            for (int j=0; j<n; j++) {  
                B[i][j]=Integer.parseInt(args[c]);  
                c=c+1;  
            }  
        }  
        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: -

C:\Users\pavan\Desktop>javac Matrixaddition.java

C:\Users\pavan\Desktop>java Matrixaddition 3 0 1 2 3 4 5 2 4 7 6 8 2 9 3 4 1 4
7 3 4 6 5

6 9 4

12 7 9

3 8 14

2. Write a java program for Matrix Multiplication.

```

public class Matrixmultiplication {
public static void main (String args []) {
int n=Integer.parseInt(args [0]);
int c=1;
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]=Integer.parseInt(args[c]);
c=c+1;
}
}
}

```

```

    }
    for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
    B[i][j]=Integer.parseInt(args[c]);
    c=c+1;
    }
    }
    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: -

C:\Users\pavan\Desktop>javac Matrixmultiplication.java

C:\Users\pavan\Desktop>java Matrixmultiplication 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.

```
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 Mo=new MethodOverloading ();  
        System.out.println(Mo. add (10,11));  
        System.out.println(Mo.add (11,12,13));  
    }  
}
```

Output: -

C:\Users\pavan\Desktop >javac MethodOverloading.java

C:\Users\pavan\Desktop >java MethodOverloading

21

36

4. Write a java program to create a class Point with two data members x & y.

Include all constructors and display ().

```
import java. Lang.*;
```

```
class Trail
```

```
{
```

```
    private int x,y;
```

```
    public Trail ()
```

```
{
```

```
        x=10;
```

```
        y=20;
```

```
}  
public Trail (int r)  
{  
    x=r;  
    y=20;  
}  
public Trail (int a, int b)  
{  
    x=a;  
    y=b;  
}  
public int add ()  
{  
    return x+y;  
}  
public int sub ()  
{  
    return Math.abs(x-y);  
}  
public int multi ()  
{  
    return x*y;  
}  
}  
class Show  
{  
    public static void main (String args [])  
    {
```

```

Trail t1=new Trail ();
System.out.println("Constructor with default values");
System.out.println("add: "+t1.add () +" sub: "+t1.sub () +" multiplication:
"+t1.multi());
Trail t2=new Trail (15);
System.out.println("Constructor with one defined value");
System.out.println("add: "+t2.add () +" sub: "+t2.sub () +" multiplication:
"+t2.multi());
Trail t3=new Trail (10,15);
System.out.println("Constructor with defined values");
System.out.println("add: "+t3.add () +" sub: "+t3.sub () +" multiplication:
"+t3.multi());} }

```

Output: -

C:\Users\pavan\Desktop >javac Show.java

C:\Users\pavan\Desktop >java Show

Constructor with default values

add: 30 sub: 10 multiplication: 200

Constructor with one defined value

add: 35 sub: 5 multiplication: 300

Constructor with defined values

add: 25 sub: 5 multiplication: 150

5. Write a java program using static method.

```

class Static {
    static int square (int x) {
        return x*x;
    }
}

```

```
public static void main (String args []) {  
int c=Integer.parseInt(args [0]);  
int result=Static. square(c);  
System.out.println(result);  
} }
```

Output: -

```
C:\Users\pavan\Desktop>javac Static.java
```

```
C:\Users\pavan\Desktop>java Static 9
```

```
81
```

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.

They are the following:

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.

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.

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.

4. if-else-if Statement:

if-else-if statement is used when we need to check multiple conditions. In this statement we have only one “if” and one “else”, however we can have multiple “else if”. It is also known as **if else if ladder**.

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.

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

Ans: - we can use the switch statement as a substitute for long if...else...if ladders. The use of switch statements makes our code more readable.

The syntax of the switch statement is:

```
switch (variable/expression) { case  
value1:  
    // statements of case1  
break; case value2:  
    // statements of case2  
break;  
    .. ..  
    .. .. default:  
    // default statements  
}
```

3. Write the difference between break and continue statement

break statement	continue statement
It terminates the execution of the remaining iteration of the loop.	It terminates only the current iteration of the loop.

break resumes the control of the program to the end of loop enclosing that 'break'.	continue resumes the control of the program to the next iteration of that loop enclosing continue.
It causes early termination of a loop.	It causes early execution of the next iteration.
break stops the continuation of the loop.	continue do not stop the continuation of the loop, it only stops the current iteration.
the break can be used with switch, label.	continue cannot be executed with switch and labels.

4.What is looping statement?

Loops are used to execute a set of statements repeatedly until a particular condition is satisfied. In Java we have three types of basic loops: for, while and do-while. **for loop:**

for loop provides a concise way of writing the loop structure. Unlike a while loop, a for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

Syntax:

```
for (initialization; condition; increment/decrement)
```

```
{
```

```
statement(s);
```

```
} while
```

loop:

A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

Syntax:

```
while (boolean condition)
```

```
{ loop
```

```
statements...
```

```
}
```

do-while loop:

do while loop is similar to while loop with only difference that it checks for condition after executing the statements, and therefore is an example of Exit Control Loop.

Syntax:

```
do {  
statements...  
} while  
(condition);
```

5. Write the difference between while and do-while

while	do-while
In while loop condition is checked First and then the loop is executed.	In do-while loop the body is executed And then the condition is executed.
Loop will be executed only if the given condition is true.	Loop will be execute at least once irrespective of the condition.
No semicolon at the end of while. while(condition)	Semicolon at the end of while. while(condition);
while loop is entry-controlled loop.	do-while loop is exit controlled loop.

6. What is array? How it is created?

An array is a collection of similar types of data. It is a container that holds data (values) of one single type. For example, you can create an array that can hold 100 values of int type.

Creation of array: datatype [] array Name; datatype - it can be primitive data types like int, char, double, byte, etc. or Java objects. array Name - it is an identifier.

7. What is class?

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- Fields ◦ Methods ◦ Constructors ◦ Blocks
- Nested class and interface

8. What is constructor?

Constructor is a block of code that initializes the newly created object. A constructor resembles an instance method in java but it's not a method as it doesn't have a return type. In short constructor and method are different (More on this at the end of this guide). People often refer constructor as special type of method in Java.

Constructor has same name as the class name.

```
public class MyClass{  
  
    //This is the constructor  
  
    MyClass(){  
  
    }  
  
    ..  
  
}
```

9. What is the use of copy constructor?

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?

Use of this keyword:

“this” keyword refers to the current object in a method or constructor.

The most common use of “this” keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter) "**this**"

keyword can also be used to:

- > Invoke current class constructor
- > Invoke current class method
- > Return the current class object
- > Pass an argument in the method call
- > Pass an argument in the constructor call

11. What is method overloading?

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

If we have to perform only one operation, having same name of the methods increases the readability of the program.

Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as a(int,int) for two parameters, and b(int,int,int) for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.

12. What is static variable?

Static Variables:

When a variable is declared as static, then a single copy of the variable is created and shared among all objects at a class level. Static variables are, essentially, global variables. All instances of the class share the same static variable.

- > We can create static variables at class-level only.

>static block and static variables are executed in order they are present in a program.

13. What is access modifier?

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:

Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

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.

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.

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.

There are many non-access modifiers, such as static, abstract, synchronized, native, volatile, transient, etc.

14. Write the difference between instance and static methods.

Instance methods	Static methods
Instance methods are used to perform repetitive tasks like reading records from file, reading records from the dbms etc	Static methods are used to perform single operation like opening files, obtaining database connection etc.

Method definition does not require a static keyword to start.	Method definition must start with static keyword.
Each and every instance method must be accessed with respective object name.	Each and every instance method must be accessed with respective class name.
Result of instance method is not shared. Every object has its own copy instance method.	Result of static method is always shared by objects of same class.

15. What is object? How it is created?

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible). The example of an intangible object is the banking system.

An object has three characteristics:

State: represents the data (value) of an object.

Behavior: represents the behavior (functionality) of an object such as deposit, withdraw, etc.

Identity: An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

Creating an Object:

As mentioned previously, 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. Eg:- object ob=new object();