Assignment-1

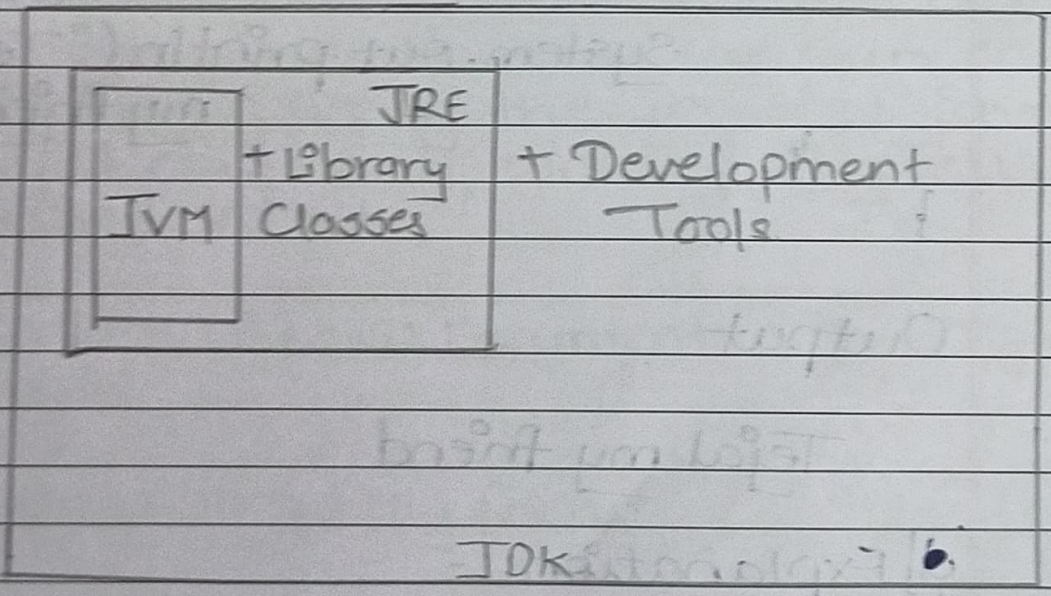
Q.1] Write a short on Java Development Kit ?

→ i] Core Java is a part of the Java programming language that one can use for developing or creating a general-purpose app.

ii] Java Development Kit is a software development environment that offers a collection of tools and libraries necessary for developing Java applications.

iii] It is a core package used in Java along with JVM and JRE

iv] JDK can be explained with the help of Daigram



v] JDK also contains the following
a] Java Runtime Environment (JRE)
b] An interpreter / loader
c] An Compiler (Javac)
d] An archiver (jar) and many more.
vi] Compiling and Running Java code using JDK.
a] We can use the JDK compiler to convert our text file into an executable program.
b] Our Java text segment is converted into bytecode after compilation which carries the .class extension
c] For example :-

```
class Tejal{
    public static void main(String[]
                                args){
        System.out.println("Tejal
                        my friend");
    }
}
```

Output

Tejal my friend.

d] Explanation :-
i] After completing code simply we should use the javac command which is used for compilation purpos

-e. And we should provide the full path of our java text file to the command line or else we will get an error as "The system cannot find the path specified" Our command should be similar to the given below example where Tejal is the file name and the full path to the file is specified before the file name. The path and javac.exe should be inside the quotes.
For example :-

```
"C:\Program Files\Java\jdk-11.0.9\bin\
    javac.exe" Tejal.java
```

Now, we can notice that the Tejal.class file is created in same directory as Tejal.java. Now we can run code by simply using the Tejal.java Tejal command, which will give us the desired result

```
C:\Users\Pinaki\Documents> java Tejal
friend
```

Output Tejal my friend.

2] List and explain the salient features of Java.

→ A] The following are the salient features of JAVA

a] Object-Oriented - i] It means that the software is organize as a combin -ation of different types of objects that incorporate both data and behaviour
ii] It is a methodology that simplifies the software development and maintenance by providing some rules. The Basic Concepts of OOPs are Object, class, inheritance, polymorphism, Abstraction & encapsulation

b] Simple - i] Java is very easy to learn & its syntax is simple clean & easy to understand
ii] Java has removed many complicated & rarely-used features Ex - explict pointers, operator overloading.

c] Secured - i] Java is best known for its security. With Java we can develop virus-free systems
ii] Java is secured because no explicit pointer, Java programs inside a virtual machine sandbox, class load -er, Byte code verifier, security Manager.

d] Platform

d] Platform independent - i] Java is platform independent because it is different from other languages like C, C++, etc which are compiled into platforms specific machines while java is a write once, run anywhere language

e] Robust (strong) - i] Java is robust because
i] It uses strong memory managem -ent
ii] There is lack of pointers that avoids security problems
iii] Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of object which are not being used by a Java application anymore
iv] There are exception handling and the type checking mechanism in Java.
v] All these points make java robust

f] Portable - i] Java is Portable because it facilitates you to carry the Java bytecode to any platform
ii] It doesn't or require any implementation

g] Architecture-neutral - i] Java is Architectural neutral because there are no implementation dependent features, for. eg - the size of primitive type is fixed

h) Dynamic - i) Java is dynamic language
ii) It supports the dynamic loading of classes
iii) It means classes are loaded on demand
iv) It also supports function from its native languages ie C and C++
v) Java supports dynamic compilation and automation memory management.
i) Multi-Threaded - i) A thread is like a separate program, executing concurrently
ii) Java Programs that deal with many tasks at once by defining multiple threads
iii) The main advantage of multi threading is that it doesn't occupy memory for each thread
j) Distributed - i) Java is distributed because it facilitates users to create distributed application in Java
ii) RMI and EJB are used for creating distributed applications
k) High-Performance - i) Java is faster than other traditional interpreted programming languages because Java bytecode is close to native code
ii) It is still a little bit slower than a compiled language
iii) Java an interpreted language that is why it is slower than compiled language.

Q] List and explain compenents of Java Virtual Machine.

→ i] Java Virtual Machine is a Virtual Machine that run Java Program that are also compiled to Java bytecode
ii] The Original JVM was designed as an interpreter for bytecode.
iii] The components of Java Virtual Machine are
a] Classloader
i] Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program it is loaded by the class loader. There are three built-in classloaders in Java.
i] Bookstrap classloader
ii] Extension classloader
iii] Application classloader
b] Class(Method) Area
i] Class (Method) Area stores per-class structure such as the runtime constant pool, field and method data the code for methods
c] ~~Heap~~ Heap.
i] Is the runtime data area in which objects are allocated
d] stack
i] Java stack store frames
ii] It holds local variables and partial results and Plays a part in method

innovation and return.
iii) A new frame is created each time a method is invoked.
iv) A frame is destroyed when its method innovation completes.

e) Program Counter Register
i) PC (Program Counter) register contains the address of the java virtual machine instruction currently being executed
f) Native Method stack
i) It contain all the Native method used in the application
g) Execution Engine
i) It contains A virtual processor interpreter and Just-In-Time compiler
ii) It is used to improve the performance
iii) If it compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.

h) Java Native Interface
i) Java Native Interface is a framework which provides an interface to communicate with another application written in another language
ii) Java uses JNI framework to send output to the console or interact with OS libraries

4) Write in detail about different types of operators in Java, category wise quoting their functionality, operands and return type. Give one example statement for each.

→ i) Java provides a rich operator environment.
ii) The Java has various types of Operator
a) Arithmetic operator
b) Bitwise Operator
c) Relational Operator
d) Logical Operator
e) Assignment Operator
f) Ternary Operator
g) Shift Operator.
iii) These above Operators are explained further in detail

a) Arithmetic Operator - The Arithmetic Operators are used to perform simple Mathematical Operations. These Operator includes +, −, *, /, %, ++, −−

i) '+' = 'A+' operator is used for Addition operation
ii) '−' = '−' operator is used for Subtraction purpose
iii) '*' = '*' operator is used to find product of two numbes.
iv) '/' = '/' is used to divide
v) '%' = '%' Modulus operator is used

Program.

```
Public class Operators
    public static void main(String[] args)
    { int a = 10;
      int b = 20;
      System.out.println(a+b);
      System.out.println(a-b);
      System.out.println(a*b);
      System.out.println(a/b);
      System.out.println(a%b);
    }
```

Output.

```
30
-10
200
0.5
0
```

Functionality

b) Bitwise Operator - Bitwise operator are the operator used for manipul -ation of individual bits of number. Bitwise operator includes the followi -g

i) ~ is called as Bitwise unary NOT operator & is used for inverting all of the bits of its operand

ii) & - & Bitwise AND operator is used for the purpose that it produces a 1 bit if both operands are also 1. A zero is produced in all other cases.

iii) | - Bitwise OR combines bits such that if either of the bits in the operands is 1, then resultant bit is 1.

iv) Eg -

```
import java.io.*;
class GfG {
Public static void main (String[] args)
{ int d=011010, e=0b1100
System.out.println,
```

c) Unary Operators - Unary operators need only operators that are used to increment, decrement or negate a value

i) - : Unary minus used for negating the values.

ii) + : Unary plus indicates the positive value if performs automatic conversion to int when the type of operated is byte, character, short

iii) ++ - increment operator is used to increments value by 1.

iv) -- - decrement operator is used for decrement value by 1.

v) ! = logical not operator used for inverting a boolean value

Eg -

```
import java
    import java.io.*; // C
    class GfG {
      Public static void main(String[] args
    { int a=10;
      System.out.println("Post increment: "+
                         (a++));
      System.out.println("Pre increment: "+(++a
      System.out.println("Post decrement: "+(a--
      System.out.println("Pre decrement: "+(--a
    }
}
```

O/P

```
Post increment = 10
pre increment = 10
Post decrement = 10
Pre decrement = 8
```

d] Assignment operator - Assignment Operator used to assign value to to any variable. It has right to left associativity

i] += - It is used for adding left operator with right operand and two assigning it to the variable on the left

ii] -= : It is used for ~~mu~~ subtracting the right operand from left operand and then assigning the variable on left

iii] *= : It is used for multiplying left operand with the right operand and then assigning the variable on left.

iv] /= It is used for dividing left operand with the right operand

v] %= for assigning the modulo of the left operand by right operand

vi] for example -

```
import java.io.*;
class GFG {
Public static void main (string[] args)
{ int a=7;
system.out.println("a+=3:"+(a+=3));
system.out.println("a-=2:"+(a-=2));
system.out.println("a*=4:"+(a*=4));
system.out.println("a/=3:"+(a/=3));
System.out.println("a%=2:"+(a%=2));
}
}
```

Output

```
a+=3:10
a=2.5
a*=4:28
a/=3:2.4
a%=2:0.14.
```

e] shift operator - It is used to shift all the bit values to the particular side of a specified number of time.

```
eg - public class n {
    public static void main (string args[])
    {
    System.out.println(10<<0);
    System.out.println(10<<3);
    System.out.println(20>>>2);
    system.out.println(20>>3);
```

Output

```
40
80
5
2
```

F] Relational Operator
- They are use to check relationship between two operands.

i] == - check if values are equal

ii] != - check if two values are not equal

iii] > : check if left operand is greater than right

iv] < : check if right operand is greater than left

v] >= : Check if left operand is greater than equal to right operand

vi] <= : check if right operand is greater than equal to left operand

for example.

```
import java.io.*;
public class A
  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 <= b" + (a <= b));
  System.out.println("a > b: " + (a > b));
  System.out.println("a < b: " + (a < b));
  }.
```

Output

a == b: false
a != b: true
a >= b: false
a <= b: true
a > b: false
a < b: true

Date
Page

56 What are the primitive data types in Java? Briefly explain their size, range and other details.

⇒ i) Primitive data types are the most basic data types that are built into Programming Language.
ii) They are used to store simple values such as numbers or characters
iii) Java has eight primitive data types which are as follows.
a) byte.
b) short
c) int
d) long
e) char
f) float
g) double
h) boolean
iv) The above primitive data types has been explained in detail
a) Byte - i) The smaller integer type is known as byte.
ii) It Represents a signed 8-bit integer value, which is equal to 1 byte
iii) It ranges from -128 to 127
iv) It is often used when working with raw binary data or when memory space is limited.
v) Eg - byte b, c;

b) Short - i) Short is data type used of size 16 bits which is equal to 2 bytes.
ii) It Represents signed integer from -32,768 to 32,767
iii) It is less used in Java
iv) It is Generally used in array indexing or when memory space is a constraint
c) int - i) Int stands for Integer
ii) int has 32 bits which is equal to 4 bytes.
iii) int ranges from ~~-2,147~~ -2,147,483,648 to 2,147,483,647.
iv) int are datatypes which is most commonly used to control loops and to index arrays
v) Eg - int a;
int b;

d) long - i) long is a datatype which is useful for those occasion where an int type is not large enough to hold desired value
ii) too long is a signed 64-bit which is equal to 8 byte
iii) It ranges from -9,223,372,036,854,775,808
iv) to 9,223,372,036,854,775,807.
v) Eg - long myLong = 10000000000L;

e) Char - i) The data type used to store character is char.
ii) It's size is 16 bits which is equal 2 byte
iii) It's ranges upto 0 to 255
iv) The standard set of characters known as ASCII
v) Eg - char myChar = 'A';

f) float - i) The float datatype specifies a single-precision value that uses 32-bits of storage which is equal to 4-byte
ii) It can represent decimal values with single-precision
Eg - float myFloat = 3.14f

g) double - i) Double precision is denoted by keyword double
ii) It is a 64-bit
iii) It ranges from upto 4.9e-32
iv) Eg - double myDouble = 3.14159;

h) boolean - i) boolean datatype is used for logical values
ii) It can have one of two possible values true or false
iii) It is of size 1-bit
iv) It range is 0 or 1. 0 stand for false and 1 stands for True
v) Eg - boolean myBoolean = true;

6] Explain about memory management in java with reference to stack and heap.

→ i] Memory management is the process of allocation and de-allocation of objects.
ii] Java does memory management automatically.
iii] Java uses an automatic memory management system called a garbage collector.
iv] The Memory allocation in java is divided into parts which are Heap, Stack, Code and Static
v] Here we will see two allocation, Stack and Heap in detail

a] Stack - i] Stack is a data structure that represents LIFO (Last in First out) collection of object allowing for pushing/popping elements in constant time
ii] The class can also be said to extend vector, and treats the class as a stack with the five mentioned function
i] Iterable
ii] Collection
iii] List
iv] Vector
v] Stack
iii] The class support one default constructor stack() which is used to create an empty stack.
iv] Eg - Public class stack<E> extends Vector<E>

b] Heap - i] A heap is a chunk of memory which is shared among all threads.
ii] In a heap, all class instances and the array is allocated.
iii] Heap is divided into two parts Young Generation and Old Generation which is explained below in detail
a] Young Generation - i]
iv] For managing the memory automatically java provides the garbage collector that deletes object which are no longer being used
v] If heap space is full, it throws the java.long.outofMemoryError.

7] Explain the terms.

i] Narrowing -
ii] Widening -

i] Narrowing -

→ i] Converting a higher datatype to a lower datatype is known as narrowing.
ii] Narrowing is also known as explicit typecasting which is conversion i.e explicitly performed in following situation
a] Narrowing a wide primitive type to a smaller primitive type value
b] Narrowing a superclass reference to a subclass reference during inheritance.
iii] For Example -

```
class A
{public static void main (String ar)
{ double d = 10.5;
    byte b= (byte) d;
    short s=(short) d;
    int i= (int) d;
    float f =(float) d
    long l =(long) d;
System.out.println("double value"+d);
System.out.println("short value"+s);
System.out.println("int value"+i);
System.out.println("float value"+f);
System.out.println("long value"+l);
System.out.println("byte value"+b);
}}
```

Output

```
double value: 10.5
short value: 10
int value: 10
float value: 10.5
long value: 10
byte value: 10.
```

Widening -

→ i) Widening conversion takes place when two data type are automatically converted
ii) It is also known as implicit conversion or casting down.
iii) Widening takes place when a smaller primitive type value is automatically accomodated in a larger primitive data type
iv) Widening also takes place, when a reference variable of a subclass is automatically accomodated in a reference variable of its ~~suf~~ superclass
v) For example -

```
class A
{ public static void main (string ar)
  { byte b=10
    short s=b;
    int i=b;
    long l=b;
    float f=b;
    double d=b;
System.out.println("short value"+s);
system.out.println("int value"+i);
system.out.println("long value"+l);
system.out.println("float value"+f);
system.out.println("double value"+d);
}}
```

Output

```
short value:10
int value:10
long value:10
float value:10.0
double value:10.0
```

8] Write in clear detail about static keyword.

→ i] The static keyword in Java is used for memory management mainly
ii] We can apply static keyword with variables, methods, blocks and nested class
iii] The static keyword is used for a constant variable or a method that is the same for every instance of class
iv] Examples of static keyword with their applications are,

a] static block

```
import java
Public class block {
  static int i=10; n;
   static {
  system.out.println ("static block")
   n = i * 8;
  }
  Public static void main (string[] arg);
  }
  system.out.println("inside main");
  system.out.println(" value of i" + i);
  system.out.println("value of n" + n);
}}
```

Output

```
Static block
insid main
value of i: 10
value of n: 80.
```

b] Static Variable - When you declare variable as static then single copy of variable is created, dividing among all objects at class level.

```
Class test {
   Static int age; }
Class main {
      test.age = 20
class test {
      static int max = 10;
       int min = 5; }

Public class main {
    {test obj = new test();
    system.out.println("min+1:"+(obj.min+1);
    system.out.println("max+1:"+(tes.max+1);
}}
```

Output

```
min+1 = 6
max+1 = 11.
```

Q] Write a short note on access specifiers in java.

→ i] Access specifiers are the keywords like "Public", "Protected", "default" and "Private" which has its special meaning in Java

a] Public access specifiers.

i] "Public" is a keyword which is introduced in java.

ii] The access scope of the "Public" is everywhere like in all classes and method as well

iii] If we prefixed "Public" keyword with any class, variable or method that it can be accessed by any class or method

b] Protected access specifiers.

i] "Protected" is the keyword which is introduced in java

ii] The access scope of "Protected" is not everywhere and it is accessible in the same class or its child class or in all those classes which are defined in the same package

iii] If we prefixed "Protected" keyword with any class, variable or method then it can be accessed by the same class or its child classes or all the classes which are defined in the same package.

c] default access specifiers.

i] The access scope of the "default" is not everywhere

ii] It is not mandate to prefixed "default" keyword with any class, variable or method because by default class, variable or method is default Public in java & it can be accessed by all those classes which are defined in some package only

d] Private access specifiers.

i] The access scope of the "Private" is not everywhere

ii] If we prefixed "Private" keyword with any variable or method then it can be accessed only in the same class