VIII. Resources required

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity	Remark
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows / Linux		
3	Software	jdk1.8.0 or above.		

IX. Resources used

S. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	Computer System with broad specifications	Intel pentium 2.00 GB 64-bit O.S.		
2	Software	Windows 7 ultimate		
3	Any other resource used	Notepad		

X. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write installation directory path of your directory?
2. Write value of path environment variable?
3. List folders created after installation.
4. Main method is declared as static. Justify.
5. Program is named with class containing main method. Justify.

(Space for answer)

X

1. C:\program files\Java\

2. C:\program files\Java\jdk 1.80\bin

3. bin, include, jre, lib

4. Static is commonly used in java coding & is doesn't create object in code of Java.

5. Program saved name and class name is common because it is becomes easy to execute Java code.

XI

3.

① open notepad and write java program

② save the programe with the name same as the class name and .java ex.
i.e HelloWorld.java

③ AND save it to specific folder/Directory

④ open cmd and set path as per you have save programe there.

⑤ compile the program javac <program_name>.java
i.e javac HelloWorld.java

⑥ Run the program java <program_name>
i.e java HelloWorld

XI Exercise. G-P

1] Write the ~~replace~~ options provided by following JDK tool along with their use.

① Java

Java interpreter which ~~can~~ run App and Application by read and interpreti the ~~bit~~ byte Code into machine knowing C or languages.

② java C

The java compiler which translate java Source Code to ~~bit~~ byte Code So the interpreter can understand.

③ javadoc.

It is use to create html documtion format from java Source Code.

2] List different Version of JDK.

① JDK Beta.
② JDK 1.0.
③ JDK 1.1
④ J2 SE 1.2
⑤ J2 SE 1.3
⑥ J2 SE 1.4
⑦ J2 SE 5.0
⑧ Java SE 9 (LTS)
⑨ Java SE 10 (18.3).
⑩ Java SE 11 (18.9 LTS)
⑪ Java SE 12 (

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	Windows 7	1	
2	Any other resource	Notepad	1	

X. Program Code: Teacher must assign a separate
3-4 students.

Write any program to check multiple conditions using if statement.

```
Class If Statement
{
 Public static Void main (String args[])
 {
 Scanner S = new Scanner (System.in);
 System.out.println ("Enter the number")
 int number = S.nextInt ();
 if (number > 0)
 {
  System.out.println ("Number is Positive
  }
  System.out.println ("This statement is
                       always executed");
 }
}
```

XI. Result (Output of Code):

Enter three integer
11, 50, 21
Second number is largest.

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List operators used in if conditional statement.
2. In if-else construct which part will be executed if condition is true.
3. State the condition when the else part will be executed with ex...

X

Program to find largest number among three number using nested if.

```
import java.util.Scanner
class Largest
{
public Static Void main (String args[])
{
 int x, y, z;
System.out.println ("Enter three integer");
Scanner in = new scanner (System.in);
x = in.nextInt();
y = in.nextInt();
Z = in.nextInt();
if (x>y && x>z)
 System.out.println ("first number is largest");
else if (y>x && y>z).
System.out.println ("second number is
                    largest");
Clese if (z>x && z>y).
System.out.println ("Third number is
                    largest");
else.
system.out.println ("The numbers are not
                    distinct");
}
}
```

4. Which of the following operator is used in if:
   a. Assignment operator (=) b. comparison operator (= =)

(Space for answer)

1. If statement evaluates a condition. To do that, we use the if if statement and conditional (ternary) operator which we will be referring to as the "question mark" operator? for the simplicity and the condition is a simple equality check (year == 2015) it can be much more complex.

2. It is used to declare whether a certain statement or block of statement will be executed or not i.e if a certain condition is true the a block of statement is executed otherwise not. Here condition after evaluation will be either true or false.

1. operator used in if Conditional statement:
   a) ==
   b) !=
   c) >
   d) >=
   e) <
   f) <=

3. then when 'if' condition is false then 'else' condition part will be executed.
   Eg.
   n=10;

```
if (n==0)
{
System.out.println ("Hi");
}
else
{
System.out.println ("Hello");
}
```

Output :- Hello.

Because n=10 and 'if' condition n==0 is false therefore it will execute the 'else' part.

4. Comparision operator (==) is used in if.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	public class NestedIfExample {
	
	public static void main(String args[]){
	int num=70;
	if(num< 100){
	System.out.println("number is less than 100");
	if(num> 50){
	System.out.println("number is greater than 50");
	}
	}
	}
	}
		Number is less than 100.

2.	class IfStatement {
	public static void main(String[] args) {
	int number = 10;
	if (number > 0) {
	System.out.println("Number is positive.");
	}
	System.out.println("This statement is always executed.");
	}
	}
		Error:
		Class name, 'If Statement', are only accepted if an notation processing is explicity requested.
		1 error

2. Write a program to make the use of logical operators.
3. Write a program to check no is even or odd.

(Space for Answer)

3Ans: Class Number
{
public Static Void main (String args[])
{
int num;
num = Integer.parseInt (args[0]);
(num % 2 == 0) ? System.out.println ("Even");
System.out.println("odd");
}
}

2 Ans:

```
import java.util.Scanner;
class Largest of Three numbers
{
public static void main (String args[])
{
int x, y, z;
System.out.println("Enter three integers");
Scanner in = new Scanner (System.in);

x = in.nextInt();
y = in.nextInt();
z = in.nextInt();

If(x>y && x>z)
System.out.println("first no is largest");
else if(y>x && y>z)
System.out.println("Second no is largest");
else if(z>x && z>y)
System.out.println("Third no is largest");
else
System.out.println("Entered no are not
                    distinct");
}
}
```

```
break; // break is optional

default :
    // Statements
}
```

2. Conditional if (ternary operator):

Syntax:

```
result = testStatement.? value1 : value2;
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	Window 7	1	
2	Any other resources	Notepad.	1	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write any program using switch-case statement.

```
class Demoswitch.
{
Public Static Void main(String args[])
{
 int c;
 int a= Integer.parse Int (args[0]);
 int b = Integer.parse Int (args[1]);
 int ch = Integer.parse Int (args[2]);
Switch (ch).
{
 Case 1:
 c=a+b; System.out.println("Addition is ="+
 break;
```

XI. Result (Output of Code):

```
1.Addition                    | 12
2.Substraction                | Enter the 3rd number
3.Multiplication              | 22
4.Division                    | Enter your choice:
Enter the 1st number          | 1
                              | Addition = 34.
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must desig more such questions so as to ensure the achievement of identified CO.

1. What will happen if break is not written for a case in switch case?
2. When default case is executed?
3. List datatypes allowed in switch expression?
4. Write a program to make use of ternary operator.

(Space for Answer)

X]

```
class DemoSwitch
{
public static void main(String args[])
{
int c;
int a = Integer.parseInt(args[0]);
int b = Integer.parseInt(args[1]);
int ch = Integer.parseInt(args[2]);
switch(ch)
{
case 1:
c=a+b;
System.out.println("Addition of two nos="+c);
break;
case 2:
c=a-b;
System.out.println("Subtraction of two nos="+c);
break;
case 3:
c=a*b;
System.out.println("Multiplication of two nos="
break;
case 4:
c=a/b;
System.out.println("Division of two nos="+c);
break;
default:
System.out.println("Wrong Choice");
}
}
```

1. Switch case statement are used to execute only specific code statement based on the switch expression. If we do not use break statement of the end of each case, program will execute all consecutive case statements until it finds next break statement or till the end of switch case block.

2. When any of the case is didn't match then the default case will be executed.

3. Primitive datatype: byte, short, char and int Enumerated type (java enums) String Class, a few classes that wrap primitive types: character, byte, short & integer

4. Public class Ternary operator
{
public static void main (String args[])
{
int num1, num2;
num1 = 10
num2 = (num1 == 1) ? 20 : 30;
System.out.println("value of num is:" + num);
num2 = (num1 == 10) ? 20 : 30;
System.out.println("value of num is = " + num2);
}
}

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	public class SwitchCaseExample1 {
	public static void main(String args[]){
	int num=2;
	switch(num+2)
	{
	case 1:
	System.out.println("Case1: Value is: "+num);
	case 2:
	System.out.println("Case2: Value is: "+num);
	case 3:
	System.out.println("Case3: Value is: "+num);
	default:
	System.out.println("Default: Value is: "+num);
	}
	}
	}
		Default Value is:2

2.	public class Program {
	public static void main(String[] args) {
	int value = 100;
	switch (value) {
	case 100:
	System.out.println(true);
	break;
	case 100:
	System.out.println(true);
	break;
	}
	}
	}
		error:
		duplicate case label 1
		case 100:
		1 error

2. Write any program to check switch-case statement using character datatype.

(Space for Answer)

```
2 import java.util.*;
class Vowel
{
public static void main(String args[]
{
Scanner S = new Scanner(System.in);
System.out.println("Enter the character");
```

```
char c=s.next().charAt(0);
switch(ch)
{
  case 'A':
  case 'E':
  case 'I':
  case 'O':
  case 'U':
System.out.println("Vowel");
break;
default:
System.out.println("Consumant");
}
}
}
```

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software.	JDK 1.8.1	1	
2	Any other resource	NotePad.	1	

X. Program Code: Teacher must assign a separate program statement to group of 3. 4 students.

Develop a program to print command line argument using for loop.

```
Class display.
{
Public Static void main (String args [])
{
int num = Integer. ParseInt (args [0]);
for (num = 10; num <= 50; num++)
  {
  if (num % 2 == 0)
  {
  System. out. println ("Even No are" + num)
  }
 }
}
}
```

XI. Result (Output of Code):

12	18	24	30	36	42	
14	20	26	32	38	44	48
16	22	28	34	40	46	50

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. When for loop will be terminated?
2. Can we write a for loop without initialization? If yes, give example.
3. Write a for loop to increment index variable by 2 in each iteration.
4. When for loop will be executed infinitely?

(Space for answer)

1. In the for loop when the value of i reaches to the if condition will be true and the break statement will be executed and the loop will be terminated.

2. It happens when the loop condition is always ~~executed~~ evaluated as true. * its perfectly legal to skip any of the 3 part of the ~~loop~~ for loop. * Below give for loop will run infinite times. * To terminate this program press Ctrl+C in the console.

3.
```
class Even_Num
{
public static void main(String args[])
{
int i;
for(i=0; i<=50; i+=2)
{
System.out.println(i);
}
}
}
```

```
4. for (;;)
    System.out.println("Hello");
```

XIII. Exercise:

1. Write any program using if condition with for loop.
2. Write any program to display pyramids of stars/patterns using increment/decrement

(Space for Answer)

```
2. Class pyramids
   {
   Public Static Void main (String args[])
   {
   integer i, j, n=5;
   for (i=0; i<n; i++)
   {
```

```
for (j=n-i; j>1; j--)
{
System.out.print(" ");
}
for (j=0; j<=i; j++)
{
System.out.print("*");
}
System.out.println();
}
}
}
```

1. Class PrimeNo
```
{
Public Static Void main (String args[])
{
int num = Integer.parseInt(args[0]);
int flag=0;
for (int i=2; i<num; i++)
{
if (num%i==0) {
System.out.println(num+"is not a prime
                    no.");
flag=1;
break;
}}
if (flag==0)
System.out.println(num+"a prime no");
}
}
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	JDK 1.8.1	1	
2	Any other resource	Notepad	1	

X. Program Code: Teacher must assign a separate program statement to group of 4 students.

Develop a program to use logical operators in do-while loop.

```
Class Dowhile.
{
Public static void main (String args[])
{
int i=0;
do
{
system.out.println("i is :"+i);
i++;
}while(i<5);
}.
```

XI. Result (Output of Code):

```
Enter no :9.0
Enter no :9.0
Enter no :9
Enter no :0
Sum = 198.0
```

XI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State difference between while and do-while loop.
2. In do-while loop termination condition is checked at _____(beginning/end)
3. How many times do-while loop will be executed if condition is false?

(Space for answer)

1]

While loop	Do-while loop
① It is the entry controlled loop.	① It is an exit controlled loop.
② Condition is written at the start of the loop.	② Condition is written at the end of the loop.
③ Loop statement do not execute if the condition is false.	③ The loop statement do execute once also if condition is false.

④ Syntax :
while (condition)
{
statements;
}
	④ Syntax :
	do
	{
	statements;
	} while (condition);

2] In do-while-loop termination condition is checked at end.

3] The do-while loop will execute once also if condition is false.

III. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class DoWhileBasics
	{
	public static void main(String args[])
	{
	int a=1;
	do
	{
	System.out.println(a);
	a=a+1; // or a++;
	}
	while(a<=10);
	}
	}
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10

2.	class Test {
	public static void main(String[] args)
	{
	while(true) {
	System.out.print(1);
	do {
	System.out.print(2);
	} while (false);
	}
	}
	}
		Run time error

2. Write a program to display number 1 to 50 using do-while loop.

XIV. References/ Suggestions for Further Reading

1. https://www.codesdope.com/c-loop-and-loop/
2. https://www.youtube.com/watch?v=IIX6cLed73o
3. https://www.journaldev.com/16536/java-do-while-loop
4. https://beginnersbook.com/2015/03/do-while-loop-in-java-with-example/

(Space for answer)

```
2] class PrintNumber
{
Public Static Void main (String argS[])
{
int i=1;
do
{
System.out.println ("i+" ");
} while (i<=50);
}
}
```

Output = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 - - - - - - - - - - - - - - 50.

Following table shows the casts that result in a loss of information.

Sr. No.	From	To
1.	byte	short, char, int, long, float, double
2.	short	int, long, float, double
3.	char	int, long, float, double
4.	long	float, double
5.	float	double

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	JDK 1.8.1	1	
2	Any other resource	Notepad.	1	

X. Program Code: Teacher must assign a separate program statement to group of 4 students.

Develop a program to show the use of implicit typecasting.

```
Public class Implicit Casting.
{
Public static void main (String args[])
{
  byte i = 50;
  short j = i;
  int k = j;
  long l = k;
 float m = l;
 double n = m;
System.out.println("byte Value:"+i);
System.out.println("Short Value:"+j);
System.out.println("int Value:"+k);
System.out.println("long Value:"+l);
System.out.println("float Value:"+m);
system.out.println("double Value:"+n);
} }.
```

XI. Result (Output of Code):

byte Value : 50 float Value : 50.0
Short Value : 50 double Value : 50.5
Int Value : 50
long Value : 50

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List different data types according to storage capacity.
2. State need of typecasting.
3. State the data types to which boolean datatype is implicitly casted.
4. Write two examples of implicit type casting.

(Space for answer)

1) Byte - 1 byte
short - 2 bytes
int - 4 bytes
long - 8 bytes
float - 4 bytes
double - 8 bytes

2) Ty casting, or type conversion is a method of changing an entity from one data type to another. It is used in Computer programming to ensure variables are correctly processed by a function. An example of typecasting is converting an integer to a string.

3) • Boolean, The type whose values are either true or false
• Char, The character type whose values are 16 bit unicode characters
• The arithmetic types: The integral types: byte, Short, int, long. The floating-point

types: float, double

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class Test{
	public static void main(String[] args) {
	int i = 100;
	long l = i;
	float f = l;
	System.out.println("Int value "+i);
	System.out.println("Long value "+l);
	System.out.println("Float value "+f);
	}
	}
		int Value 100
		Long Value 100
		float Value 100.0

2.	public class Test{
	public static void main(String[] argv) {
	char ch = 'c';
	int num = 88;
	ch = num;
	}
	}
		in compatiable types! possible lossy conversion from int to char
		ch = num;

3. Write a program to implicitly typecast lower range data type to larger storage size datatype.

```
Class Parent
{
Public void display()
{
System.out.println("Parent disp called");
}
}
Public class child extends Parent.
{
Public static void main (String args[3])
{
Parent P = new child();
P.disp();
}
}
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software.	jdk 1.8.1	1	
2	Any other Resource	Notepad.	1	

X. Program Code: Teacher must assign a separate program statement to group 4 students.

Develop a program to show the use of explicit type casting.

```
Public Class EXPlicit Casting.
{
 Public Static Void main (String args[])
 {
   double d = 75.0;
   float f = (float) d;
   lang l = (lang) f;
   int i = (int) l;
   Shor S = (Short) i;
   byte b = (byte) S;
System.out.println("double value:="+d);
System.out.println(" float value:="+f);
System.out.println("lang value:="+l);
System.out.println("Short value:="+S);
System.out.println("int value:="+i);
System.out.println("byte value:="+b);
 }
}
```

XI. Result (Output of Code):

```
double value : 75.0 | int value = 75
float value : 75.0  | Short value = 75
long value : 75.0   | byte value = 75
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What is casting?
2. What is difference between implicit and explicit type casting?
3. What is narrowing?

(Space for answer)

1] well all casting really means is taking an object of one particular type and "turning it into" another object type This process is called casting a Variable. This topic is not specific to Java, as many other programming language support casting of their Variable types.

3] The Narrowing conversion: Data type conversion <<Data type << Java Tutorial: The narrowing conversion occurs from a type to a different type that has a smaller size, such as from a along (64 bits) to an int (32 bits) In general, the narrowing primitive conversion can occur in these cases... int to byte short to char.

2]

Implicit	Explicit
①Implicit conversions are done automatically by the compiler.	①Explicit conversions require programmers approval.
②widening cast	②Narrowing cast
③No data loss.	③Data may be lost.

pace.

	Error/Output
rgs) {	Double value 100.04
lue "+d);	Long value 100
e "+l);	Int value 100
+i);	
s[]) {	100

3.
```
class Test{
  public static void main(String args[])  {
    byte a= 4;
    char b = 'z';
    short c = 102;
    int i = 5000;
    float f = 5.7f;
    double d = .124;
    double result = (f * a) + (i / b) - (d * c);
    System.out.println("result = " + result);
  }
}
```

Result = 50.151999237060547

2. Write a program to convert variable of basic datatypes and shows result of explicit typecasting.

(Space for Answer)

```
class parent
{
public s Void disp()
{
System.out.println("parent disp called");
}
}

public class child extends parent
{
public Void disp()
{
System.out.println("child disp called");
}
public Static Void main (String args[])
{
Parent p = new child ();
p.disp();
Child c = p;
c.disp();
}
}
```

class_name

{

Statements for initialize the data members

}

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software.	jdK 1.81	1	
2	Any other Resource	Notepad.	1	

X. Program Code: Teacher must assign a separate program statement to group of 4 students.

Demonstrate use of at least two types of constructors.

```
class Language.
{
String name;
Language().
{
System.out.println("Constructor
method called");
}
Language(String t)
{
name = t;
}
Public Static Void main(String
                        args[])
{
 Language CPP= new language();
Language Java= new language
                        ("Java");
CPP.set name("C++");
Java.get Name();
CPP.get Name();
}
Void set Name(Strin
{
 name = t;
}
 Void get Name()
{
System.out. Println(
 guage name" +nam
}
}
```

XI. **Result (Output of Code):**

Constructor method called
language name java
language name c++

XII. **Practical Related Questions**

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Does constructor return a value?
2. Specify the situation when the default constructor is provided by the system.
3. Specify the situation when the default constructor is explicitly defined in the class.
4. How constructor overloading can be done?

(Space for answer)

1] No Constructor does not return a value. yet it returns a value is correct sentence because it is not called directly by your code, it's called by the memory allocation and object initialization that code in the run-time, So constructor cannot have a return type. But hald on! it returns something.

2] If you don't implement any constructor in your class, the Java compiler inserts default constructor into your code on your behalf. You will not see the default constructor in your source code as it is inserted during compilation & present in the byte code.

3] A default constructor is automatically created only when you do not define any constructor yourself.
If you need two constructor one with arguments & one withoub

you need to an usually define both.

4] In addition to overloading methods, we can also overload constructors in Java. overloaded constructor is called based upon the parameter specified when new is executed. Sometimes there is a need of initializing an object in different ways. This can be done using constructor overloading.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	class T {
	int t;
	}
	class Main {
	public static void main(String args[])
	{
	T t1 = new T();
	System.out.println(t1.t);
	}
	}
		0

2. Modify the following program to execute without error. State which constructors are used in the program.

```
class Point
{
  int m_x, m_y;
  public Point(int x, int y)
  {  m_x = x;   m_y = y;  }
 public static void main(String args[])
 {
     Point p1 = new Point();
     Point p = new Point(2,3);
    System.out.println("X"+p.m_x);
    System.out.println("Y"+p.m_y);
    System.out.println("X"+p1.m_x);
    System.out.println("Y"+p1.m_y);
 }
}
```

3. Write a program to implement different types of constructors to perform addition of complex numbers.

(Space for Answer)

```
Class Complex
{
int Real, Image;
ComplexC() { } Complex C int Real1, int Image) {
Real = Real1; Image = Image1; }
Complex AddComplex Ccomplex c1, Complex c2) {
Complex Csum = new Complex C);
Csum.Real = c1.Real + c2.Real;
Csum.Image = c1.Imag + c2.Image;
return Csum; }}
Class ComplexIn {
```

```
Public static void main (String [] a) {
Complex C1
Complex C2
Complex C3
C3 = C3.Add complex {C1, C2};
System.out.println("Sum:" + C3.Real" + i" + C3.Image)
}
}
```

XIV. References/ Suggestions for Further Reading

1. https://www.youtube.com/watch?v=lrYghXs9EEU
2. https://freevideolectures.com/course/2513/java-programming/17

XV. Assessment Scheme

9.	int indexOf(int ch)	string
10.	String substring(int startIndex)	A method returns last occurrence of character value or substring.
11.	int lastIndexOf(int ch)	

String Buffer class methods:

Sr. No.	Syntax	Task Performed
1.	StringBuffer append (StringBuffer sb)	Appends specified StringBuffer with StringBuffer
2.	StringBuffer insert (int offset, String str)	This method inserts a string str at po mentioned by offset.
3.	void setLength(int newlength)	Sets the length of the character sequ
4.	void setCharAt(int index, char ch)	The character at specified index of t StringBuffer is set to ch.
5.	StringBuffer reverse()	Reverse the character sequence in th StringBuffer.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Rema
1	Software	jdk 1.8.1	1	
2	Any other resource	Notepad.	1	

X. **Program Code: Teacher must assign a separate program statement to group of 3-4 students.**
Write a program to show the use of all methods of String class.

```
class StringMethods
{
Public static void main (String [] args) {
String str = "Java is fun to learn";
String S1 = "Java";
String S2 = "JAVA";
String S3 = " Hello Java";
System.out.println("Char at indexe 2:"+Str.charAt(2));
System.out.println("After concat:"+str.concat("-Enjoy."));
System.out.println("ignoring case:"+S2.equals IgnoreCase
                                        (S1));
system.out.println("checking equal with case:"S2.equals
                                        (S1));
system.out.println("checking length:"+ Strlength());
system.out.println("Replace string:"+str.replace
                                ("fun", "easy"));
system.out.println("substring:"+Str.substring(8));
System.out.println("substring:"+str.substring(8,12));
System.out.println("lower case:"+str.toLowerCase())
System.out.println("Upper case:"+str.toUpperCase());
System.out.println("Triming String:"+S3.trim());
system.out.println("Searching S1:"+Str.contains(S1));
system.out.println("Searching S2:"+Str.contains(S2));
Char[] CharArray = S2.toCharArray();
System.out.println("size of char array:"+CharArray.
                                        length);
System.out.println("Printing last element of array:"+
                                CharArray[3]);
}
}
```

XI. Result (Output of Code):

Trimming String : "Hello Java"
Searching S1 : false
Searching S2 : True
Size of char array : 4
Printing length element at array : 9

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must de more such questions so as to ensure the achievement of identified CO.

1. List different constructors of String class along with syntax
2. List different constructors of StringBuffer class along with syntax.
3. State whether String is primitive datatype or class in Java? State the package
4. What is difference between ==, equals() and compareTo() method?

(Space for answer)

1] uppercase - S1. to Upper Case ();
Lowercase - S1. to Lower Case ();
Length S1. Length ();
Replace - S1. replace ('x', 'y');
Trim - S1. Trim ();
Char At - S1. Char At ();
Equals - S1. equals (S2);
Compare - S1. compare To (S2);
Index of - S1. index of ('x', n);

2] Set Char - S1. Set char At ('x', 'n');
Append - S1. append (S2);
Insert - S1. insert (n, S2);
Set Length - S1. set length (n);
Reverse - S1. reverse ();
Delete - S1. delete (n, m);
replace - S1. replace (n, m, 'x');

3] A string is class in Java
Packages : A java package is group of similar types of classes, interfaces and sub packages. Package in java can be

categorized in two form, built-in package and user-designed package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

4] Another difference between them is that, if both "==" and equals() is used to compare objects. Than == returns true only if both references points to same object while equals() can return true or false based on its overridden implementation. one of the popular cases is comparing two string in java in which case ==.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	class String_demo{
	public static void main(String args[]) {
	char chars[] = {'a','b','c'};
	String s = new String(chars);
	System.out.println(s);
	}
	}
		abc.

2.	class Output{
	public static void main(String args[]) {
	String s1 = "Hello I love Java" ;
	String s2 = new String(s1);
	System.out.println((s1==s2) +" " +
	s1.equals(s2));
	}
	}
		false true.

2. Write a program to implement all methods of StringBuffer class.

(Space for Answer)

```
class StringBufferExample
{
public static void main(String args[])
{
StringBuffer sb=new StringBuffer("Hello");
sb.append("Java");
System.out.println(sb);
sb.replace(1,3,"Java");
System.out.println(sb);
sb.insert(1,"Java");
System.out.println(sb);

sb.delete(1,3);
System.out.println(sb);
sb.reverse();
System.out.println(sb);
System.out.println(sb.capacity());
}
}
```

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software.	jdK 1.8.1	1	
2	Any other resource	NotePad.	1.	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students:

Write a program to implement multidimensional array.

```
class ThreeDmatrix.
{
Public Static Void main
        (String [] args)
{
int threeD [][][] = new
        int [3][4][5];
int i,j,k;
for (i = 0; i<3; i++)
for (j=0; j<4; j++)
for (K=0; K<5; K++)
threeD [i] [j] [K] =i*j*K;
 for (i =0; i<3; i++) {.
 for (j= 0; j<4; j++) {
 for (K=0; K<5; K++)
  {
System.out.println (three
 D [i] [j] [K]);
 }
system.out.println ();
 }
system.out.println ();
 }
 }
}.
```

XI. Result (Output of Code):

00000	0 0000	00000
00000	01234	02486
00000	02468	0481216
00000	03 6 912	06 1 1 18 24

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What is use of new operator in defining an array?
2. In 2D array which dimension is optional at the declaration of array?
3. Is it possible to change size of array once allocated?
4. State the situation where Index Out of Bounds Exception will be generated.

(Space for answer)

1] An array is an instance of a special Java array class and has a corresponding type in the type system. This means that to use any array as with any other object, we first declare a variable of the appropriate type of Then we use the new operator to create and instance of it

3] There's no easy way to automatically resize an array as its contents expend you can call realloc on an array you've allocated on the heap to make it larger but ideally the resizing of the array would be automatic

4] The Array IndexOut of Bounds Exception is a Runtime Exception thrown only at runtime. The Java Compiler does not check for This error during the compilation of a program.

XIII. Exercise:

1. Write output/error of code in the given space.

Sr. No.	Program Code	Output/Error

1.	State line no and error.
	classTest2 {
	public static void main(String[] args) {
	inta[] = new int[5]; // line 1
	int[] arr = new int[]; // line 2
	}
	}
		array dimension missing.

2.	class Test5 {
	public static void main(String[] args) {
	int arr[] = new int[5];
	System.out.println(arr);
	System.out.println(arr[0]);
	}
	}
		[I@15db9742

3. Write a program to display array elements using for-each loop.

```
import java.util.Arrays;
Public Class Array Example(Space for answer)
{
Private int[] intArray = new int[] {1,2,3,4,5};
Private String[] strArray = new String[] {"abc",
"bcd", "def", "efg");
Public Static Void main(String args[]) {
System.out.println("Print int array in java :"+int
                                        Array);
System.out.println("Print String array in java :"
                                        +strArray);
for(int i=0; i<intArray.length; i++) {
System.out.println(intArray(i) + ", ");}
for(int i : intArray) {
```

System.out.println(i+", "); }
System.out.println("print string array is Java: "+Array.
asList(strArray));
System.out.println("print int Array value in Java: "+Arrays.
asList(intArray));
System.out.println("Print values of array in java"+Array.toString
(intArray));
System.out.println("Print values of array in java"+Array.toString
(strArray));
int[][]twoDimenArray=new int[][]{{1,2,3},{10,20,30},{100,200,300},
{60,53}};
System.out.Println("print two dimensional array "+Array.deepToString
(twoDimenArray)); 33.

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/array-in-java
2. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
3. https://www.youtube.com/watch?v=okHL1h5rhNM

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		**70%**
1	Logic formation	30%
2	Debugging ability	30%
3	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4	Expected output	10%

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	JdK 1.8.1	1	
2	Any other resource	Notepad	1	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to insert different elements in the Vector and display them.

```
import java.util.Vector;
Public class vectorDemo
{
Public static void main (String
                    args[])
{
Vector<Integer> vec= new vector
                <Integer>(4);
  vec.add(4);
  vec.add(3);
  vec.add(2);
  vec.add(1);
Vector system.out.println
("Added number are:");
for (Integer number=vec):
{
System.out.println("Number="
                +number);
}
Vec.insertElementAt(133,1);
System.out.println("Added
number after insertion:");
for (Integer number:vec)
{
System.out.println("Number
                ="+number);
}
}
}
```

XI. Result (Output of Code):

Added number one:	Added number after insertion
number = 4	number = 4
number = 3	number = 133
number = 2	number = 3
number = 1	number = 2
	number = 1

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State difference between size() and capacity() method of Vector class.
2. Differentiate between addElement() and insertElementAt() methods of Vector class.
3. Differentiate between array and Vector.

(Space for answer)

→ 1) capacity() and size() of Vector in java: The difference between capacity() and size() in java.util.Vector is that the size() is the number of elements which is currently hold and capacity() in the number of element which can maximum hold. A vector defaults to doubling the size of its Array.

→ 3]

Array	Vector
① work faster.	① works comparatively slower.
② Non-Synchronized.	② Synchronized.
③ A non-Synchronized data structure that uses a dynamic array for storing the element.	③ A synchronized data structure that uses a dynamic array for storing the element.
④ use Iterator interface to traverse the element.	④ uses the Interator Interface or Enumeration interface to traverse the element.

2] The java.util.Vector addElement() method is used to append a specified element to the end of this Vector by increasing the size of the vector by 1. The functionality of this method is similar to that of the add() method of vector class.

Vector insertElementAt() method in java The Java.util.Vector.insertElementAt(element, index) method is used to insert a particular element at the specified index of the Vector.

XIII. Exercise:

Sr. No.	Program Code	Output

1.	import java.util.*;
	class demo1 {
	public static void main(String[] args)
	{
	Vector v = new Vector(20);
	System.out.println(v.capacity());
	System.out.println(v.size());
	}
	}
		20
		0

1. Write a program to use different methods of Vector class.

(Space for answer)

```
import java.util.*;
Class VectorDemo
{public Static void main(String args[]) {
Vector V = new Vector(3,2);
System.out.println("Initial Size: "+V.size());
System.out.println("Initial Capacity: "+V.capacity());
V.addElement(new Integer(3));
V.addElement(new Double[5.45]);
V.addElement(new float(1,4));
V.addElement(new character('N'));
V.add Element(new string("Naresh"));
V.add Element(new Integer(1));
```

```
System.out.println(v);
System.out.println("Current capacity :"+v.capacity());
System.out.println("first element:"+(Integer)v.firstElement());
System.out.println("last element :"+(Integer)v.lastElement());
if(v.contains(new Integer(3))) System.out.println("Vector contain 3.");
Enumeration venum=v.element(); System.out.println("\nElement in Vector");
while(v.Enum.hasmoreElement()) System.out.println(v.Enum.nextElement()+" "); System.out.println();
```

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/array-in-java
2. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
3. https://www.youtube.com/watch?v=okHL1h5rhNM

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		70%
1	Logic formation	30%
2	Debugging ability	30%

Similar Wrapper class methods are available for Float, Short, Long and Double class.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software	JdK 1.8.1	1	
2	Any other resource.	Notepad	1	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of Integer Wrapper class methods.

```
import java.lang.*;
Public class wrapperDemo.
{
 Public static void main (String args[])
{
 Integer intobj1 = new Integer (25);
 Integer intobj2 = new Integer ("25");
 Integer intobj3 = new Integer (35);
 String s = "123";
System.out.println ("obj1 = "+intobj1+"obj2= "+
intobj2+"obj3 = "+intobj3);
 Integer intobj = Integer.valueOf(s);
 int i = obj.intValue();
 i += 2;
System.out.println(i);
}
```

XI. 0
-1

XII. **Practical Related Questions**

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write a different ways to create object of the any primitive datatype.
2. Write methods of Number class to convert object into primitive datatypes.
3. List all Wrapper classes in Java.

(Space for answer)

3] ① Boolean
② Character
③ Byte
④ Short
⑤ Integer
⑥ Long
⑦ float
⑧ Double

1] a] using new keyword. This is the most common way to create an object in Java.
Sy : Myobject object = new Myobject();
b] using class for Name(). If we know the name of the class and if it has a public default construction we can create an object in this way.
Sy : Myobject object = (Myobject) Class.forName("...")
c] using clone(); The clone() can be used to create a copy of an existing object.
Sy : Myobject anotherobject = new Myobject
Myobject Object = (Myobject) another object.clone();

d] Using object deserilization object deserialization is nothing but creating an object from its Serialized form.

```
Sy: ObjectInputStream inStream = new ObjectInputStream(CanInputstream)
My object object = (My object)instream.readobject();
```

XIII. Exercise:

1. Write a program to convert String value into Integer Wrapper class object.
2. Write a program to make use of Character Wrapper class methods.
3. Write a program to convert Integer object value into primitive datatype byte, short and double value.

(Space for answer)

1]

```
Public class MyString To Integer
{
  public static void main (String a[])
  {
    String str = "23";
    Integer i = Integer.Valueof (Str);
    System.out.println("The integer value:"+i);
  }
}
```

2]

```
public class Test
{
  public static Void main (string [] args)
  {
System.out.println(character.tostring ('x'));
System.out.println(character. to string ('Y'));
system.out.println(character.toLowerCase ('A'));
```

```
System.out.println(Character.toLowerCase(65));
System.out.println(Character.isDigit('5'));
System.out.println(Character.isDigit('-'));
System.out.println(Character.toUpperCase('a'));
System.out.println(Character.toUpperCase(97));
}
}
```

XIV. References/ Suggestions for Further Reading

1. https://www.youtube.com/watch?v=IM3c6el6lO8
2. https://www.tutorialspoint.com/java/java_numbers.htm

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		70%
1.	Logic formation	30%
2.	Debugging ability	30%
3.	Follow ethical practices	10%

3] Write a program to convert Integer object value into primitive datatype byte, short and double value.

→ public class MyIntegerByteValue.

```
{
public Static Void main (String a[])
{
Integer itr = new Integer (10);
system.out.println ("byte value"; +itr.byteValue());
system.out.println ("double value :" +itr.doubleValue ());
System.out.println ("Float Value:" +itr.FloatValue ());
System.out.println ("int value :" + itr.intValue ());
System.out.println ("long value:" +itr.longValue());
System.out.println ("Short Value:" +itr.ShortValue());
}
}
```

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Software.	JDK 1.8.1	1	
2	Any Other resource	Notepad	1	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate the use of Overriding method display( ) using Super and Sub classes

Note: Attach the code at the end.

```
class Super
{
int x;
Super(int x)
{
this.x=x;
}
Void display()   //Method defined
{
System.out.println("super x ="+x);
}
}
```

```
class Sub extends Super
{
  int y;
  Sub(int x, int y)
  {
    super(x);
    this.y = y;
  }
  void display()
  {
    System.out.println("Super x = "+x);
    System.out.println("Sub y = "+y);
  }
}
class override Test
{
  public static void main(String args[])
  {
    Sub S1 = new Sub(100, 200);
    S1.display();
  }
}
```

XI. Result (Output of Code):

Super x = 100
Sub y = 200

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State the difference between method overloading and method overriding.
2. Method Overriding is an example of.....(Compile Time Polymorphism/Run Time Polymorphism)
3. Write the rules of method overriding.
4. Write the use of super keyword in method overriding.

(Space for answer)

Overloading	Overriding
1. Overloading happens at compile-time. The binding OR of overloaded method call to its definition has happens at Compile-time.	1. Overriding happens at runtime binding of overriding method call to its definition happens at runtime.

2. polymorphism applies not to overloading	2. Polymorphism applies to overriding
3. overloading is a compile time concept	3. overriding is a run time concept

2] Method overriding is an example of compile time polymorphism.

3] Rules for method overriding:

* in Java a method can only be written in subclass not in same class.
* The argument list should be exactly the same as that of the overridden method.
* The return type should be the same or a subtype of the return type declared in the original overriden
* static cannot be overriden but can be re-declared.
* If a method cannot be inherited then it cannot be overriden.

4] ① Supper can be used to refer immediate parent class instance variable.

② Super can be used to invoke immediate parent class method.

③ Super() can be used to invoke immediate parent class constructor.

```
class MyChildClass
{
  public void disp()
  {
    System.out.println("Child class method");
  }
  public static void main (String args[])
  {
    MyChildClass obj = new MyChildClass();
    obj.disp(); }}
```

3. Develop a program to extend 'dog' from 'animal' to override 'move( )' method using super keyword.

(Space for answer)

```
1) class Bank
{
 int getRate of Interest()
 {
   return 0;
 }}
 class SBI extends Bank
 {
 int getRate of Interest()
 {
 return 8;
 }}
 Class ICICI extends Bank
 {
 int getRate of Interest()
 {
 return 7; }}
```

```
class Test2 {
public static Void main (string args [ ]) {
SBI S = new SBI ( );
ICICI I = new ICICI ( );
System.out Println("SBI Rate of Interest: "+S.getRateofInte
                                                    ( ));
System.out.Println("ICICI Rate of Interest: "+i.get Ratcof
                                          Interest ( ));
}
}
```

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/method-overriding-in-
2. https://www.tutorialspoint.com/java/java_overriding.htm
3. Programming with Java by E. Balagurusamy
4. https://beginnersbook.com/2014/01/method-overriding-in-java-with-example/

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		70%
1.	Logic formation	30%
2.	Debugging ability	30%
3.	Follow ethical practices	

3] Develop a program to extend 'dog' from 'animal' to override 'Move()' Method using Super Keyword.

→
```
class Animal
{
public Void move()
{
System.out.println("Animals can move");
}
}
class Dog extends Animal
{
public Void move()
{
Super.move();
System.out.println("Dogs can walk and run");
}
}
public class Test Dog
{
public static Void main(String args[])
{
Animal b = new Dog();
a.move();
}
}
```

The keyword extends indicates that the properties of the superclassname are extended to the subclassname.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	software	jdk 18.1	1	
2	any other resource	Notepad	1	

Multilevel inheritance

```
class X
{
 public Void methodX()
 {
 System.out.println("class X method");
 }
}
Class Y extends X
{
 public Void methodY()
 {
 System.out.println("class Y method");
 }
}
class Z extends Y
{
 public Void methodz()
 {
 System.out.println("class z method");
 }
 public Static Void main(String args[])
 {
  Z obj = new z();
  obj.methodX();
  Obj.methodY();
  obj.methodz();
 }
}
```

Program Code: Teacher must assign a separate program to each group of 4 students.

Write a program to implement single and multilevel inheritance.

Note: Attach the code at the end.

Single Inheritance.

```
public class Shape.

int length;
int breadth;

public class Rectangle extends Shape.

int area;
public Void CalculateArea().
{
area = length * breadth;
}
Public Stadic Void main(String args[])
{
Rectangle r = new Rectangle();
r.length = 10;
r.breadth = 20;
r.CalculateArea();
System.out.println("The area of rectangle
of length\" "+r.length+"\" and breadth\""
+r.breadth+"\" is \""+r.area+"\"");
}.
}
```

XI. Result (Output of Code):

①The area of Rectangle of length 10 and breadth 20 is 200

② class x method

class y method.

XII. Practical Related Questions

Note: Below given are few sample questions for reference... more such questions so as to ensure the achievement of identified CO.

1. Justify: Java does not support multiple inheritance.
2. Specify the conditions when the super keyword can be used.
3. Write the importance of final variables and methods.
4. Specify the conditions which needs to be satisfied while using the abstract classes.

(Space for answer)

1] inheritance Suppose C is the child class extending from both parent class a and parent class b with same methods defined in them Then child class cannot understand which class method to call. So there is a confusion here which leads to ambiguity and leads to compile time error. This is the reason Java does not support multiple Inheritance.

2] It is used inside a Subclass method definition to call a method defined in the super class. private methods of the superclass cannot be called: only public and protected methods can be called by the super keyword. [2] It is also used by class constructors to invoke constructors of its parent class. [3] super Keyword are not used in static method.

3] *Final Variable.

1] When the variable is declare as final intialize the Value of it then you cannot change the value of that particular Variable.

2] Declaring a final variable globally and initializing it's Value in constructor is possible.

* Final method

1. A method declared as final can be overloaded.
2. A method declared as final, can be overloaded in inheritance also (from one class to another class).

XIII. Exercise:

1 Develop a program to implement the multilevel inheritance.

2 Develop a program to calculate the room area and volume to illustrate the concept of single inheritance (Assume suitable data wherever necessary).

(Space for answer)

1]
```
class X
{
public void methodX() {
System.out.println("class X method");
}}
```

```
class Y extends X
{
public void method Y() {
System.out.println("class Y method");
}}
class Z extends Y
{
public void method Z() {
System.out.println("class Z method");
}
public static void main (String args[])
{
Z obj = new Z();
obj.method X();
obj.method Y();
obj.method Z();
}}
```

2]
```
class Room
{
int length, width;
Room (int a, int b)
{
length = a;
width = b;
}
Void area() {
int area = length * breadth;
System.out.println("The area of the room is" area
}}
class roomVol extends Room
{
int height;
roomVol (int a, int b, int c)
{
```

```
int Volume = length * width * height;
System.out.println("The Volume of the room is"
                                    +Volume);
 }
}
class inheritance3
{
public static void main(String args[])
 {
  roomVol room2 = new roomVol(10, 40, 20);
  room2.area();
  room2.Volume();
 }
}
```

super(a,b);
height = c;
}
void Volume()
{

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/inheritance-in-j
2. https://www.geeksforgeeks.org/output-java
3. https://beginnersbook.com/2013/05/java-m

XV. Assessment Scheme

X. **Program Code: Teacher must assign a separate program statement to group of 3-4 students.**
Demonstrate the use of interfaces to implement the concept of multiple inheritance.
Note: Attach the code at the end.

```
interface X
{
Public Void mymethod();
}
interface Y
{
Public Void myMethod();
}
class JavaExample implements X, Y
{
Public Void myMethod()
 {
 System.out.println("Implementing more than
                     one interfaces");
 }
 Public static Void main(String args[])
 {
  JavaExample obj = new JavaExample();
  Obj.MyMethod();
 }
}.
```

XI. Result (Output of Code):

Implementing more than one interface.

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Differentiate between Class and Interface.
2. Write similarities between interfaces and classes.
3. Write advantages of interfaces.
4. Will this code compile successfully?(Assume Suitable Class)

```
public interface FamousLine
{
    void ShowLine( )
    {
    System.out.println("Show Line");
    }
}
```

(Space for answer)

(1) class	Interface
a] support only multi-level and hierarchical inheritances but not multiple inheritance.	a] supports all types of inheritance multi-level hierarchical & multiple inheritance.
b] can have main() method	b] cannot have main() method as main() is a concrete method.
c] can have constructor	c] cannot have constructor
d] Variable can be private	d] Variables should be public only.
e] Methods can be final & static.	e] Methods should not be final & static.

2] Similarities are:
a] They are both Java basic object types
b] They both can contain variables and methods (with difference being class members have implementation code whereas the inheritance interface methods code where only have declarations)
c] They can both be inherited using inheritance (extends keyword for classes & implements keyword for interface.

3] Advantages of interfaces:
1] Through interfaces we can implement multiple inheritance in Java.
2] Interface function to breakup the complex designs and clear the dependencies between objects.

4] No, This code will not compile successfully because in interface we only declare method we can't define it so, This will not compile.

```
}
class Dog implements Pet
{
    public void test()
    {
        System.out.println("Interface Method Implemented");
    }
    public static void main(String args[]){
    Pet p = new Dog();
    p.test();
    }
```

(Space for answer)

2] Interface Area
{
final static float pi = 3.14F;

```
float compute(float x, float y);
}
class Rectangle implements Area
{
public float compute(float x, float y)
{
return (x * y);
}
}
class circle implements Area
{
public float compute(float x, float y)
{
return (pi * x * x);
}
}
class InterfaceTest
{
public static void main(String args[])
{
Rectangle rect = new Rectangle();
circle cir = new circle();
System.out.println("Area of Rectangle="
+rect.compute(10, 20));
System.out.println("Area of circle =" +cir.
compute(10, 0));
}
}
```

1) → Error : NewCircle is not abstract and does not override abstract method draw() in NewShape.

```
class NewCircle implements circle
```

3) → Output : Interface method Implemented.

XIV. References/ Suggestions for Further Reading

1. https://www.geeksforgeeks.org/interfaces-in-java/
2. Programming with java by E Balagurusamy

XV. Assessment Scheme

Performance Indicators		Weightage
Process related (35 Marks)		**70%**
1.	Logic formation	30%
2.	Debugging ability	30%
3.	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4.	Expected output	10%
5.	Timely Submission	10%

IX.

Sr. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	Softwar	jdK 18.1	1	
	any other resource	Notepad	1	

X. **Program Code: Teacher must assign a separate program statement to group of 3-4 students.**

Write a program to implement user defined packages in terms of creating a new package and importing the same.

Note: Attach the code at the end.

```
Package myPack1;
Public classA
{
 Public Void show()
 {
System.out.println
("ClassA");
}
}
```

```
Package myPack2;
import myPack1.A;
class B
{
Public static void main
(String args[]).
{
A.classA=new A();
  ClassA.show();
 }
}
```

XI. **Result (Output of Code):**

class A

XII. **Practical Related Questions**

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Name some of java packages.
2. Can we import same class/package twice? Will the JVM load the package twice at run time?
3. Write fully qualified and shortcut class naming approach with examples.

(Space for answer)

1] 1 Java.lang

2. Java.io

3. Java.util

4. Java.applet

5. Java.awt

6. Java.net

2] one can import the same package or same class multiple time. Neither compiler nor JVM complains about it and the JVM will internally load the class only once no matter how many times you import the same class

3] In a computer programming, a fully qualified naming is an unambiguous name that specifies which object, function or variable a call refers to without regarded to the context of the call

XIII

① Package let_me_calculator;
public class calculator
{
public void add (int a, int b)
{
int c = a+b;
System.out.println ("sum" + c);
}
}
import java.util.*;
import let_me_calculate.calculator;
class Demo
{
~~Scanner~~ s public static void main (String args[])
{
Scanner s = new Scanner (System.in);
System.out.println ("enter two number");
int n1 = s.nextInt ();
int n2 = s.nextInt ();
calculator c = new calculator ();
c.add (n1, n2)
}
}

2. Define a package named myInstitute include class named as department with one method to display the staff of that department. Develop a program to import this package in a java application and call the method defined in the package.
3. Develop a program which consists of the package named let_me_calculate with a class named calculator and a method named add to add two integer numbers. Import let_me_calculate package in another program (class named Demo) to add two numbers.

(Space for answer)

2]
```
Package myInstitue;
public class Department
{
    public int deptno;
    public String dept name, Staff name,
                  Subject name;
public void display (int dno, String dname,
  String staffn, String Sname)
{
    deptno = dno;
    deptname = staffn; dname; Staffname = sta
    subject_name = Sname;
System.out.println ("Department No: "+ deptno)
System.out.println ("Department Name: "+ deptnam
System.out.println ("Staffname: "+ Staffname);
System.out.println ("Subject Name: "+ Subjectnam
  }
}
```

```
package mypack2;
import mypack1.A;
class B
{public static void main(String args[])
 {A classA = new A();
  classA.show();
 }
}
```

XIV. References/ Suggestions for Further Reading

1. https://beginnersbook.com/2013/03/packages-in-java/
2. https://www.javatpoint.com/package
3. Programming with java by E Balagurusamy

XV. Assessment Scheme

Performance Indicators	Weightage
lated (35 Marks)	70%
	30%