



Zeal Education Society's
ZEAL POLYTECHNIC, PUNE.

NARHE | PUNE -41 | INDIA

SECOND YEAR (SY)

DIPLOMA IN COMPUTER ENGINEERING

SCHEME: I

SEMESTER: IV

NAME OF SUBJECT: JAVA PROGRAMMING

SUBJECT CODE: 22412

MSBTE QUESTION PAPERS & MODEL ANSWERS

- 1. MSBTE SUMMER -19 EXAMINATION**
- 2. MSBTE WINTER -19 EXAMINATION**

22412

21819

3 Hours / 70 Marks

Seat No.

--	--	--	--	--	--	--	--

- Instructions* –
- (1) All Questions are *Compulsory*.
 - (2) Answer each next main Question on a new page.
 - (3) Figures to the right indicate full marks.
 - (4) Assume suitable data, if necessary.
 - (5) Use of Non-programmable Electronic Pocket Calculator is permissible.
 - (6) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

Marks

- 1. Attempt any FIVE of the following:** **10**
- a) List any eight features of Java.
 - b) State use of finalize() method with its syntax.
 - c) Name the wrapper class methods for the following:
 - (i) To convert string objects to primitive int.
 - (ii) To convert primitive int to string objects.
 - d) List the types of inheritances in Java.
 - e) Write the syntax of try-catch-finally blocks.
 - f) Give the syntax of < param > tag to pass parameters to an applet.
 - g) Define stream class. List its types.

P.T.O.

- 2. Attempt any THREE of the following:** **12**
- a) Explain the concept of platform independence and portability with respect to Java language.
 - b) Explain the types of constructors in Java with suitable example.
 - c) Explain the two ways of creating threads in Java.
 - d) Distinguish between Input stream class and output stream class.
- 3. Attempt any THREE of the following:** **12**
- a) Define a class student with int id and string name as data members and a method void setData (). Accept and display the data for five students.
 - b) Explain dynamic method dispatch in Java with suitable example.
 - c) Describe the use of following methods:
 - (i) Drawoval ()
 - (ii) getFont ()
 - (iii) drawRect ()
 - (iv) getFamily ()
 - d) Write a program to count number of words from a text file using stream classes.
- 4. Attempt any THREE of the following:** **12**
- a) Describe instance Of and dot (·) operators in Java with suitable example.
 - b) Explain the four access specifiers in Java.
 - c) Differentiate between method overloading and method overriding.
 - d) Differentiate between Java Applet and Java Application (any four points)
 - e) Write a program to copy content of one file to another file.

5. Attempt any TWO of the following:**12**

- a) Describe the use of any methods of vector class with their syntax.
- b) Explain the concept of Dynamic method dispatch with suitable example.
- c) Write a program to create two threads. One thread will display the numbers from 1 to 50 (ascending order) and other thread will display numbers from 50 to 1 (descending order).

6. Attempt any TWO of the following:**12**

- a) Explain the command line arguments with suitable example.
 - b) Write a program to input name and salary of employee and throw user defined exception if entered salary is negative.
 - c) Describe the applet life cycle in detail.
-



SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

Important Instructions to examiners:

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No	Sub Q.N.	Answer	Marking Scheme
1.	a) Ans.	Attempt any <u>FIVE</u> of the following: List any eight features of Java. Features of Java: 1. Data Abstraction and Encapsulation 2. Inheritance 3. Polymorphism 4. Platform independence 5. Portability 6. Robust 7. Supports multithreading 8. Supports distributed applications 9. Secure 10. Architectural neutral 11. Dynamic	10 2M <i>Any eight features 2M</i>
	b) Ans.	State use of finalize() method with its syntax. Use of finalize(): Sometimes an object will need to perform some action when it is	2M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<p>destroyed. Eg. If an object holding some non java resources such as file handle or window character font, then before the object is garbage collected these resources should be freed. To handle such situations java provide a mechanism called finalization. In finalization, specific actions that are to be done when an object is garbage collected can be defined. To add finalizer to a class define the finalize() method. The java run-time calls this method whenever it is about to recycle an object.</p> <p>Syntax: protected void finalize() { } }</p>	<p><i>Use 1M</i></p> <p><i>Syntax 1M</i></p>
	<p>c) Ans.</p>	<p>Name the wrapper class methods for the following: (i) To convert string objects to primitive int. (ii) To convert primitive int to string objects. (i) To convert string objects to primitive int: String str="5"; int value = Integer.parseInt(str); (ii) To convert primitive int to string objects: int value=5; String str=Integer.toString(value);</p>	<p>2M</p> <p><i>1M for each method</i></p>
	<p>d) Ans.</p>	<p>List the types of inheritances in Java. <i>(Note: Any four types shall be considered)</i> Types of inheritances in Java: i. Single level inheritance ii. Multilevel inheritance iii. Hierarchical inheritance iv. Multiple inheritance v. Hybrid inheritance</p>	<p>2M</p> <p><i>Any four types ½M each</i></p>
	<p>e) Ans.</p>	<p>Write the syntax of try-catch-finally blocks. try{ //Statements to be monitored for any exception } catch(ThrowableInstance1 obj) { //Statements to execute if this type of exception occurs } catch(ThrowableInstance2 obj2) { //Statements }finally{</p>	<p>2M</p> <p><i>Correct syntax 2M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		//Statements which should be executed even if any exception happens }	
f) Ans.	Give the syntax of < param > tag to pass parameters to an applet. Syntax: <param name="name" value="value"> Example: <param name="color" value="red">	2M <i>Correct syntax 2M</i>	
g) Ans.	Define stream class. List its types. Definition of stream class: An I/O Stream represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays. Streams support many different kinds of data, including simple bytes, primitive data types, localized characters, and objects. Java's stream based I/O is built upon four abstract classes: InputStream, OutputStream, Reader, Writer. Types of stream classes: i. Byte stream classes ii. Character stream classes.	2M <i>Definitio n 1M</i> <i>Types 1M</i>	
2. a) Ans.	Attempt any <u>THREE</u> of the following: Explain the concept of platform independence and portability with respect to Java language. <i>(Note: Any other relevant diagram shall be considered).</i> Java is a platform independent language. This is possible because when a java program is compiled, an intermediate code called the byte code is obtained rather than the machine code. Byte code is a highly optimized set of instructions designed to be executed by the JVM which is the interpreter for the byte code. Byte code is not a machine specific code. Byte code is a universal code and can be moved anywhere to any platform. Therefore java is portable, as it can be carried to any platform. JVM is a virtual machine which exists inside the computer memory and is a simulated computer within a computer which does all the functions of a computer. Only the JVM needs to be implemented for each platform. Although the details of the JVM will defer from platform to platform, all interpret the same	12 4M <i>Explana tion 3M</i>	



SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<p>byte code.</p> <pre> graph TD SC[Source Code] --> JC[Java Compiler] JC --> BC[Byte code] BC --> JVM1[Java Virtual Machine JVM] BC --> JVM2[Java Virtual Machine JVM] JVM1 --> WOS[Window Operating System] JVM2 --> LOS[Linux Operating System] </pre>	<p><i>Diagram</i> 1M</p>
<p>b)</p> <p>Ans.</p>	<p>Explain the types of constructors in Java with suitable example. <i>(Note: Any two types shall be considered).</i></p> <p>Constructors are used to initialize an object as soon as it is created. Every time an object is created using the 'new' keyword, a constructor is invoked. If no constructor is defined in a class, java compiler creates a default constructor. Constructors are similar to methods but with to differences, constructor has the same name as that of the class and it does not return any value.</p> <p>The types of constructors are:</p> <ol style="list-style-type: none"> 1. Default constructor 2. Constructor with no arguments 3. Parameterized constructor 4. Copy constructor <p>1. Default constructor: Java automatically creates default constructor if there is no default or parameterized constructor written by user. Default constructor in Java initializes member data variable to default values (numeric values are initialized as 0, Boolean is initialized as false and references are initialized as null).</p> <pre> class test1 { int i; boolean b; byte bt; float ft; String s; </pre>	<p>4M</p> <p style="text-align: center;"><i>Explana tion of the two types of construc tors</i> 2M</p> <p style="text-align: center;"><i>Example</i> 2M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

<pre>public static void main(String args[]) { test1 t = new test1(); // default constructor is called. System.out.println(t.i); System.out.println(t.s); System.out.println(t.b); System.out.println(t.bt); System.out.println(t.ft); } }</pre> <p>2. Constructor with no arguments: Such constructors does not have any parameters. All the objects created using this type of constructors has the same values for its datamembers. Eg: class Student { int roll_no; String name; Student() { roll_no = 50; name="ABC"; } void display() { System.out.println("Roll no is: "+roll_no); System.out.println("Name is : "+name); } public static void main(String a[]) { Student s = new Student(); s.display(); } }</p> <p>3. Parametrized constructor: Such constructor consists of parameters. Such constructors can be used to create different objects with datamembers having different values. class Student { int roll_no; String name; Student(int r, String n) { roll_no = r;</p>	
--	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<pre>name=n; } void display() { System.out.println("Roll no is: "+roll_no); System.out.println("Name is : "+name); } public static void main(String a[]) { Student s = new Student(20,"ABC"); s.display(); } }</pre> <p>4. Copy Constructor : A copy constructor is a constructor that creates a new object using an existing object of the same class and initializes each instance variable of newly created object with corresponding instance variables of the existing object passed as argument. This constructor takes a single argument whose type is that of the class containing the constructor.</p> <pre>class Rectangle { int length; int breadth; Rectangle(int l, int b) { length = l; breadth= b; } //copy constructor Rectangle(Rectangle obj) { length = obj.length; breadth= obj.breadth; } public static void main(String[] args) { Rectangle r1= new Rectangle(5,6); Rectangle r2= new Rectangle(r1); System.out.println("Area of First Rectangle : "+ (r1.length*r1.breadth));</pre>	
--	--	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<pre>System.out.println("Area of First Second Rectangle : "+ (r1.length*r1.breadth)); } }</pre>	
	<p>c) Ans.</p>	<p>Explain the two ways of creating threads in Java. Thread is a independent path of execution within a program. There are two ways to create a thread: 1. By extending the Thread class. Thread class provide constructors and methods to create and perform operations on a thread. This class implements the Runnable interface. When we extend the class Thread, we need to implement the method run(). Once we create an object, we can call the start() of the thread class for executing the method run(). Eg: class MyThread extends Thread { public void run() { for(int i = 1;i<=20;i++) { System.out.println(i); } } public static void main(String a[]) { MyThread t = new MyThread(); t.start(); } } a. By implementing the runnable interface. Runnable interface has only on one method- run(). Eg: class MyThread implements Runnable { public void run() { for(int i = 1;i<=20;i++) { System.out.println(i); } } } public static void main(String a[]) { MyThread m = new MyThread(); Thread t = new Thread(m); t.start(); } }</p>	<p>4M</p> <p><i>2M each for explaining of two types with example</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		}																						
d) Ans.	<p>Distinguish between Input stream class and output stream class. Java I/O (Input and Output) is used to process the input and produce the output. Java uses the concept of a stream to make I/O operation fast. The java.io package contains all the classes required for input and output operations. A stream is a sequence of data. In Java, a stream is composed of bytes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Sr. No.</th> <th style="width: 40%;">Input stream class</th> <th style="width: 50%;">Output stream class</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Java application uses an input stream to read data from a source;</td> <td>Java application uses an output stream to write data to a destination;</td> </tr> <tr> <td style="text-align: center;">2</td> <td>It may read from a file, an array, peripheral device or socket</td> <td>It may be a write to file, an array, peripheral device or socket</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Input stream classes reads data as bytes</td> <td>Output stream classes writes data as bytes</td> </tr> <tr> <td style="text-align: center;">4</td> <td>Super class is the abstract inputStream class</td> <td>Super class is the abstract OutputStream class</td> </tr> <tr> <td style="text-align: center;">5</td> <td>Methods: public int read() throws IOException public int available() throws IOException public void close() throws IOException</td> <td>Methods: public void write(int b) throws IOException public void write(byte[] b) throws IOException public void flush() throws IOException public void close() throws IOException</td> </tr> <tr> <td style="text-align: center;">6</td> <td>The different subclasses of Input Stream are: File Input stream, Byte Array Input Stream, Filter Input Stream, Piped Input Stream, Object Input Stream, DataInputStream.</td> <td>The different sub classes of Output Stream class are: File Output Stream, Byte Array Output Stream , Filter output Stream, Piped Output Stream, Object Output Stream, DataOutputStream</td> </tr> </tbody> </table>		Sr. No.	Input stream class	Output stream class	1	Java application uses an input stream to read data from a source;	Java application uses an output stream to write data to a destination;	2	It may read from a file, an array, peripheral device or socket	It may be a write to file, an array, peripheral device or socket	3	Input stream classes reads data as bytes	Output stream classes writes data as bytes	4	Super class is the abstract inputStream class	Super class is the abstract OutputStream class	5	Methods: public int read() throws IOException public int available() throws IOException public void close() throws IOException	Methods: public void write(int b) throws IOException public void write(byte[] b) throws IOException public void flush() throws IOException public void close() throws IOException	6	The different subclasses of Input Stream are: File Input stream, Byte Array Input Stream, Filter Input Stream, Piped Input Stream, Object Input Stream, DataInputStream.	The different sub classes of Output Stream class are: File Output Stream, Byte Array Output Stream , Filter output Stream, Piped Output Stream, Object Output Stream, DataOutputStream	<p>4M</p> <p><i>Any four points for input stream class and output stream class 1M each</i></p>
Sr. No.	Input stream class	Output stream class																						
1	Java application uses an input stream to read data from a source;	Java application uses an output stream to write data to a destination;																						
2	It may read from a file, an array, peripheral device or socket	It may be a write to file, an array, peripheral device or socket																						
3	Input stream classes reads data as bytes	Output stream classes writes data as bytes																						
4	Super class is the abstract inputStream class	Super class is the abstract OutputStream class																						
5	Methods: public int read() throws IOException public int available() throws IOException public void close() throws IOException	Methods: public void write(int b) throws IOException public void write(byte[] b) throws IOException public void flush() throws IOException public void close() throws IOException																						
6	The different subclasses of Input Stream are: File Input stream, Byte Array Input Stream, Filter Input Stream, Piped Input Stream, Object Input Stream, DataInputStream.	The different sub classes of Output Stream class are: File Output Stream, Byte Array Output Stream , Filter output Stream, Piped Output Stream, Object Output Stream, DataOutputStream																						



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<pre> for(i=0;i<5;i++) { arr[i].display(); } } }</pre>	
b) Ans.	<p>Explain dynamic method dispatch in Java with suitable example.</p> <p>Dynamic method dispatch is the mechanism by which a call to an overridden method is resolved at run time, rather than compile time.</p> <ul style="list-style-type: none">• When an overridden method is called through a superclass reference, Java determines which version (superclass/subclasses) of that method is to be executed based upon the type of the object being referred to at the time the call occurs. Thus, this determination is made at run time.• At run-time, it depends on the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed• A superclass reference variable can refer to a subclass object. This is also known as upcasting. Java uses this fact to resolve calls to overridden methods at run time. <p>Therefore, if a superclass contains a method that is overridden by a subclass, then when different types of objects are referred to through a superclass reference variable, different versions of the method are executed. Here is an example that illustrates dynamic method dispatch:</p> <pre>// A Java program to illustrate Dynamic Method // Dispatch using hierarchical inheritance class A { void m1() { System.out.println("Inside A's m1 method"); } } class B extends A { // overriding m1() void m1()</pre>	<p>4M</p> <p><i>Explanation 2M</i></p> <p><i>Example 2M</i></p>	



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<pre>{ System.out.println("Inside B's m1 method"); } class C extends A { // overriding m1() void m1() { System.out.println("Inside C's m1 method"); } } // Driver class class Dispatch { public static void main(String args[]) { // object of type A A a = new A(); // object of type B B b = new B(); // object of type C C c = new C(); // obtain a reference of type A A ref; // ref refers to an A object ref = a; // calling A's version of m1() ref.m1(); // now ref refers to a B object ref = b;</pre>	
--	---	--



SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<pre>// calling B's version of m1() ref.m1(); // now ref refers to a C object ref = c; // calling C's version of m1() ref.m1(); } }</pre>	
	<p>c)</p> <p>Ans.</p>	<p>Describe the use of following methods:</p> <p>(i) Drawoval () (ii) getFont () (iii) drawRect () (iv) getFamily ()</p> <p>(i) Drawoval (): Drawing Ellipses and circles: To draw an Ellipses or circles used drawOval() method can be used. Syntax: void drawOval(int top, int left, int width, int height) The ellipse is drawn within a bounding rectangle whose upper-left corner is specified by top and left and whose width and height are specified by width and height. To draw a circle or filled circle, specify the same width and height.</p> <p><i>Example:</i> g.drawOval(10,10,50,50);</p> <p>(ii) getFont (): It is a method of Graphics class used to get the font property Font f = g.getFont(); String fontName = f.getName(); Where g is a Graphics class object and fontName is string containing name of the current font.</p> <p>(iii) drawRect (): The drawRect() method display an outlined rectangle. Syntax: void drawRect(int top,int left,int width,int height) The upper-left corner of the Rectangle is at top and left. The dimension of the Rectangle is specified by width and height. <i>Example:</i> g.drawRect(10,10,60,50);</p>	<p>4M</p> <p><i>Each method 1M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		(iv) getFamily () : The getfamily() method Returns the family of the font. String family = f.getFamily(); Where f is an object of Font class	
	d) Ans.	Write a program to count number of words from a text file using stream classes. <i>(Note : Any other relevant logic shall be considered)</i> import java.io.*; public class FileWordCount { public static void main(String are[]) throws IOException { File f1 = new File("input.txt"); int wc=0; FileReader fr = new FileReader (f1); int c=0; try { while(c!=-1) { c=fr.read(); if(c==(char)' ') wc++; } System.out.println("Number of words :"+(wc+1)); } finally { if(fr!=null) fr.close(); } } }	4M <i>Correct program 4M</i>
4.	a) Ans.	Attempt any <u>THREE</u> of the following: Describe instance Of and dot (.) operators in Java with suitable example. Instance of operator: The java instance of operator is used to test whether the object is an instance of the specified type (class or subclass or interface).	12 4M



SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<p>The instance of in java is also known as type comparison operator because it compares the instance with type. It returns either true or false. If we apply the instance of operator with any variable that has null value, it returns false.</p> <p><i>Example</i></p> <pre>class Simple1{ public static void main(String args[]){ Simple1 s=new Simple1(); System.out.println(s instanceof Simple1);//true } }</pre> <p>dot (.) operator:</p> <p>The dot operator, also known as separator or period used to separate a variable or method from a reference variable. Only static variables or methods can be accessed using class name. Code that is outside the object's class must use an object reference or expression, followed by the dot (.) operator, followed by a simple field name.</p> <p><i>Example</i></p> <p>this.name="john"; where name is a instance variable referenced by 'this' keyword</p> <p>c.getdata(); where getdata() is a method invoked on object 'c'.</p>	<p><i>Description and example of each operator</i></p> <p>2M</p>
<p>b) Ans.</p>	<p>Explain the four access specifiers in Java.</p> <p>There are 4 types of java access modifiers:</p> <p>1. private 2. default 3. Protected 4. public</p> <p>1) private access modifier: The private access modifier is accessible only within class.</p> <p>2) default access specifier: If you don't specify any access control specifier, it is default, i.e. it becomes implicit public and it is accessible within the program.</p> <p>3) protected access specifier: The protected access specifier is accessible within package and outside the package but through inheritance only.</p> <p>4) public access specifier: The public access specifier is accessible everywhere. It has the widest scope among all other modifiers.</p>	<p>4M</p> <p><i>Each access specifiers</i></p> <p>1M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

c)	Ans.	Differentiate between method overloading and method overriding.		4M <i>Any four points 1M each</i>	
		Sr. No.	Method overloading		Method overriding
		1	Overloading occurs when two or more methods in one class have the same method name but different parameters.		Overriding means having two methods with the same method name and parameters (i.e., method signature)
		2	In contrast, reference type determines which overloaded method will be used at compile time.		The real object type in the run-time, not the reference variable's type, determines which overridden method is used at runtime
		3	Polymorphism not applies to overloading		Polymorphism applies to overriding
4	overloading is a compile-time concept.	Overriding is a run-time concept			
d)	Ans.	Differentiate between Java Applet and Java Application (any four points)		4M <i>Any four points 1M each</i>	
		Sr. No.	Java Applet		Java Application
		1	Applets run in web pages		Applications run on stand-alone systems.
		2	Applets are not full featured application programs.		Applications are full featured programs.
		3	Applets are the small programs.		Applications are larger programs.
		4	Applet starts execution with its init().		Application starts execution with its main ().
		5	Parameters to the applet are given in the HTML file.		Parameters to the application are given at the command prompt
		6	Applet cannot access the local file system and resources		Application can access the local file system and resources.
7	Applets are event driven	Applications are control driven.			



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<p>e) Ans.</p>	<p>Write a program to copy content of one file to another file.</p> <pre>class fileCopy { public static void main(String args[]) throws IOException { FileInputStream in= new FileInputStream("input.txt"); FileOutputStream out= new FileOutputStream("output.txt"); int c=0; try { while(c!=-1) { c=in.read(); out.write(c); } System.out.println("File copied to output.txt...."); } finally { if(in!=null) in.close(); if(out!=null) out.close(); } } }</pre>	<p>4M</p> <p><i>Correct logic 2M</i></p> <p><i>Correct Syntax 2M</i></p>
5.	<p>a) Ans.</p>	<p>Attempt any <u>TWO</u> of the following: Describe the use of any methods of vector class with their syntax. <i>(Note: Any method other than this but in vector class shall be considered for answer).</i></p> <ul style="list-style-type: none">• boolean add(Object obj)-Appends the specified element to the end of this Vector.• Boolean add(int index,Object obj)-Inserts the specified element at the specified position in this Vector.• void addElement(Object obj)-Adds the specified component to the end of this vector, increasing its size by one.• int capacity()-Returns the current capacity of this vector.• void clear()-Removes all of the elements from this vector.• Object clone()-Returns a clone of this vector.	<p>12 6M</p> <p><i>Any 6 methods with their use 1M each</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<ul style="list-style-type: none">• boolean contains(Object elem)-Tests if the specified object is a component in this vector.• void copyInto(Object[] anArray)-Copies the components of this vector into the specified array.• Object firstElement()-Returns the first component (the item at index 0) of this vector.• Object elementAt(int index)-Returns the component at the specified index.• int indexOf(Object elem)-Searches for the first occurrence of the given argument, testing for equality using the equals method.• Object lastElement()-Returns the last component of the vector.• Object insertElementAt(Object obj,int index)-Inserts the specified object as a component in this vector at the specified index.• Object remove(int index)-Removes the element at the specified position in this vector.• void removeAllElements()-Removes all components from this vector and sets its size to zero.	
<p>b)</p> <p>Ans.</p>	<p>Explain the concept of Dynamic method dispatch with suitable example.</p> <p>Method overriding is one of the ways in which Java supports Runtime Polymorphism. Dynamic method dispatch is the mechanism by which a call to an overridden method is resolved at run time, rather than compile time.</p> <p>When an overridden method is called through a superclass reference, Java determines which version (superclass/subclasses) of that method is to be executed based upon the type of the object being referred to at the time the call occurs. Thus, this determination is made at run time. At run-time, it depends on the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed</p> <p>A superclass reference variable can refer to a subclass object. This is also known as upcasting. Java uses this fact to resolve calls to overridden methods at run time.</p> <p>If a superclass contains a method that is overridden by a subclass, then when different types of objects are referred to through a superclass reference variable, different versions of the method are executed. Here is an example that illustrates dynamic method dispatch:</p>	<p>6M</p> <p><i>Explanation 3M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<pre>/ A Java program to illustrate Dynamic Method // Dispatch using hierarchical inheritance class A { void m1() { System.out.println("Inside A's m1 method"); } } class B extends A { // overriding m1() void m1() { System.out.println("Inside B's m1 method"); } } class C extends A { // overriding m1() void m1() { System.out.println("Inside C's m1 method"); } } // Driver class class Dispatch { public static void main(String args[]) { // object of type A A a = new A(); // object of type B B b = new B(); // object of type C C c = new C(); } }</pre>	<p><i>Example 3M</i></p>
--	--	---	------------------------------



SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<pre>// obtain a reference of type A A ref; // ref refers to an A object ref = a; // calling A's version of m1() ref.m1(); // now ref refers to a B object ref = b; // calling B's version of m1() ref.m1(); // now ref refers to a C object ref = c; // calling C's version of m1() ref.m1(); } }</pre> <p>Output:</p> <p>Inside A's m1 method Inside B's m1 method Inside C's m1 method</p> <p>Explanation:</p> <p>The above program creates one superclass called A and it's two subclasses B and C. These subclasses overrides m1() method.</p> <ol style="list-style-type: none">1. Inside the main() method in Dispatch class, initially objects of type A, B, and C are declared.2. A a = new A(); // object of type A3. B b = new B(); // object of type B C c = new C(); // object of type C	
--	--	---	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<p>c)</p> <p>Ans.</p> <pre>class Ascending extends Thread { public void run() { for(int i=1; i<=15;i++) { System.out.println("Ascending Thread : " + i); } } } class Descending extends Thread { public void run() { for(int i=15; i>0;i--) { System.out.println("Descending Thread : " + i); } } } public class AscendingDescending Thread { public static void main(String[] args) { Ascending a=new Ascending(); a.start(); Descending d=new Descending(); d.start(); } }</pre>	<p>6M</p> <p><i>Creation of two threads 4M</i></p> <p><i>Creating main to create and start objects of 2 threads: 2M</i></p>
6.	<p>a)</p> <p>Ans.</p> <p>Java Command Line Argument: The java command-line argument is an argument i.e. passed at the time of running the java program.</p>	<p>12</p> <p>6M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<p>The arguments passed from the console can be received in the java program and it can be used as an input. So, it provides a convenient way to check the behaviour of the program for the different values. You can pass N (1,2,3 and so on) numbers of arguments from the command prompt.</p> <p>Command Line Arguments can be used to specify configuration information while launching your application. There is no restriction on the number of java command line arguments. You can specify any number of arguments Information is passed as Strings. They are captured into the String args of your main method</p> <p>Simple example of command-line argument in java</p> <p>In this example, we are receiving only one argument and printing it. To run this java program, you must pass at least one argument from the command prompt.</p> <pre>class CommandLineExample { public static void main(String args[]){ System.out.println("Your first argument is: "+args[0]); } }</pre> <p>compile by > javac CommandLineExample.java run by > java CommandLineExample sonoo</p>	<p><i>4M for explanation</i></p> <p><i>2M for example</i></p>
<p>b) Ans.</p>	<p>Write a program to input name and salary of employee and throw user defined exception if entered salary is negative.</p> <pre>import java.io.*; class NegativeSalaryException extends Exception { public NegativeSalaryException (String str) { super(str); } } public class S1</pre>	<p>6M</p> <p><i>Extended Exception class with constructor 2M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

	<pre> { public static void main(String[] args) throws IOException { BufferedReader br= new BufferedReader(new InputStreamReader(System.in)); System.out.print("Enter Name of employee"); String name = br.readLine(); System.out.print("Enter Salary of employee"); int salary = Integer.parseInt(br.readLine()); Try { if(salary<0) throw new NegativeSalaryException("Enter Salary amount isnegative"); System.out.println("Salary is "+salary); } catch (NegativeSalaryException a) { System.out.println(a); } } } </pre>	<p style="text-align: center;"><i>Acceptin g data 1M</i></p> <p style="text-align: center;"><i>Throwin g user defining Exceptio n with try catch and throw 3M</i></p>
<p>c) Ans.</p>	<p>Describe the applet life cycle in detail.</p> <div style="text-align: center;"> <pre> graph TD Start(()) -- init() --> Born((Born)) Born -- start() --> Running((Running)) Running -- stop() --> Idle((Idle)) Idle -- start() --> Running Running -- paint() --> Running Idle -- destroy() --> Dead((Dead)) </pre> </div> <p>Below is the description of each applet life cycle method:</p> <p>init(): The init() method is the first method to execute when the applet is executed. Variable declaration and initialization operations</p>	<p style="text-align: center;">6M</p> <p style="text-align: center;"><i>2M Diagram</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Java Programming

Subject Code: 22412

		<p>are performed in this method.</p> <p>start(): The start() method contains the actual code of the applet that should run. The start() method executes immediately after the init() method. It also executes whenever the applet is restored, maximized or moving from one tab to another tab in the browser.</p> <p>stop(): The stop() method stops the execution of the applet. The stop() method executes when the applet is minimized or when moving from one tab to another in the browser.</p> <p>destroy(): The destroy() method executes when the applet window is closed or when the tab containing the webpage is closed. stop() method executes just before when destroy() method is invoked. The destroy() method removes the applet object from memory.</p> <p>paint(): The paint() method is used to redraw the output on the applet display area. The paint() method executes after the execution of start() method and whenever the applet or browser is resized.</p> <p>The method execution sequence when an applet is executed is:</p> <ul style="list-style-type: none">• init()• start()• paint() <p>The method execution sequence when an applet is closed is:</p> <ul style="list-style-type: none">• stop()• destroy()	<p>4M descripti on</p>
--	--	--	---

22412

11920

3 Hours / 70 Marks

Seat No.

--	--	--	--	--	--	--	--

- Instructions* – (1) All Questions are *Compulsory*.
(2) Answer each next main Question on a new page.
(3) Illustrate your answers with neat sketches wherever necessary.
(4) Figures to the right indicate full marks.
(5) Assume suitable data, if necessary.
(6) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

Marks

- 1. Attempt any FIVE of the following:** **10**
- Define constructor. List its types.
 - Define class and object.
 - List the methods of File Input Stream Class.
 - Define error. List types of error.
 - List any four Java API packages.
 - Define array. List its types.
 - List access specifiers in Java.

P.T.O.

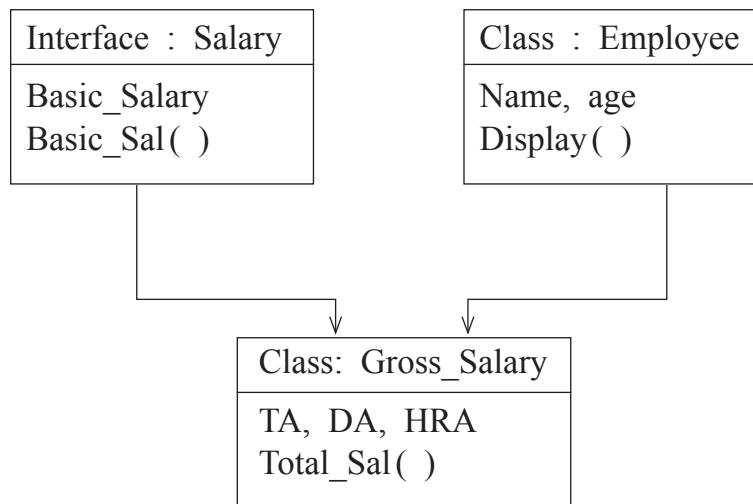
- 2. Attempt any THREE of the following:** **12**
- a) Differentiate between String and String Buffer.
 - b) Define a class circle having data members Pi and radius. Initialize and display values of data members also calculate area of circle and display it.
 - c) Define exception. State built-in exceptions.
 - d) Write a syntax and example of
 - (i) drawRect()
 - (ii) drawoval()
- 3. Attempt any THREE of the following:** **12**
- a) Explain the following classes.
 - (i) Byte Stream Class
 - (ii) Character Stream Class
 - b) Explain life cycle of Applet.
 - c) Differentiate between class and interfaces.
 - d) Define type casting. Explain its types with syntax and example.
- 4. Attempt any THREE of the following:** **12**
- a) Explain life cycle of thread.
 - b) Describe final variable and final method.
 - c) Explain any two logical operators in Java with example.
 - d) Differentiate between array and vector.
 - e) List any four methods of string class and state the use of each.

5. Attempt any TWO of the following:**12**

- a) Write a program to create vector with five elements as (5, 15, 25, 35, 45). Insert new element at 2nd position. Remove 1st and 4th element from vector.
- b) Define packages. How to create user defined package? Explain with example.
- c) Write a program to create two threads one thread will print even no. between 1 to 50 and other will print odd number between 1 to 50.

6. Attempt any TWO of the following:**12**

- a) Explain how to pass parameter to an applet? Write an applet to accept username in the form of parameter and print "Hello <username>".
- b) Write a program to perform following task.
 - (i) Create a text file and store data in it.
 - (ii) Count number of lines and words in that file.
- c) Implement the following inheritance.





MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

Winter – 19 EXAMINATION

Subject Name: Java Programming

Model Answer

Subject Code: 22412

Important Instructions to examiners:

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1.		Attempt any Five of the following:	10M
	a	Define Constructor. List its types.	2M
	Ans	Constructor: A constructor is a special member which initializes an object immediately upon creation. It has the same name as class name in which it resides and it is syntactically similar to any method. When a constructor is not defined, java executes a default constructor which initializes all numeric members to zero and other types to null or spaces. Once defined, constructor is automatically called immediately after the object is created before new operator completes. Types of constructors: 1. Default constructor 2. Parameterized constructor 3. Copy constructor	Definition: 1Mark Types: 1 Mark
	b	Define Class and Object.	2M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	Ans	Class: A class is a user defined data type which groups data members and its associated functions together. Object: It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods.	Definition 1 Mark each
	c	List the methods of File Input Stream Class.	2M
	Ans	<ul style="list-style-type: none">• void close()• int read()• int read(byte[] b)• read(byte[] b, int off, int len)• int available()	Any Two Each for 1 Mark
	d	Define error. List types of error.	2M
	Ans	<ul style="list-style-type: none">• Errors are mistakes that can make a program go wrong. Errors may be logical or may be typing mistakes. An error may produce an incorrect output or may terminate the execution of the program abruptly or even may cause the system to crash. <p>Errors are broadly classified into two categories:</p> <ol style="list-style-type: none">1. Compile time errors2. Runtime errors	Definition: 1m List: 1m
	e	List any four Java API packages.	2M
	Ans	<ol style="list-style-type: none">1.java.lang2.java.util3.java.io4.java.awt5.java.net6.java.applet	1/2 Marks for one Package
	f	Define array. List its types.	2M
	Ans	An array is a homogeneous data type where it can hold only objects of one data type. Types of Array:	Definition 1 Mark, List 1 Mark



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		1)One-Dimensional 2)Two-Dimensional													
	g	List access specifiers in Java.	2M												
	Ans	1)public 2)private 3)friendly 4)protected 5)Private Protected	Any 2, 1M for each												
2.		Attempt any Three of the following:	12M												
	a	Differentiate between String and String Buffer.	4M												
	Ans	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">String</th> <th style="width: 50%; text-align: center;">String Buffer c</th> </tr> </thead> <tbody> <tr> <td>String is a major class</td> <td>String Buffer is a peer class of String</td> </tr> <tr> <td>Length is fixed (immutable)</td> <td>Length is flexible (mutable)</td> </tr> <tr> <td>Contents of object cannot be modified</td> <td>Contents of object can be modified</td> </tr> <tr> <td>Object can be created by assigning String constants enclosed in double quotes.</td> <td>Objects can be created by calling constructor of String Buffer class using "new"</td> </tr> <tr> <td>Ex:- String s="abc";</td> <td>Ex:- StringBuffer s=new StringBuffer ("abc");</td> </tr> </tbody> </table>	String	String Buffer c	String is a major class	String Buffer is a peer class of String	Length is fixed (immutable)	Length is flexible (mutable)	Contents of object cannot be modified	Contents of object can be modified	Object can be created by assigning String constants enclosed in double quotes.	Objects can be created by calling constructor of String Buffer class using "new"	Ex:- String s="abc";	Ex:- StringBuffer s=new StringBuffer ("abc");	Any 4 Points 4 Marks
String	String Buffer c														
String is a major class	String Buffer is a peer class of String														
Length is fixed (immutable)	Length is flexible (mutable)														
Contents of object cannot be modified	Contents of object can be modified														
Object can be created by assigning String constants enclosed in double quotes.	Objects can be created by calling constructor of String Buffer class using "new"														
Ex:- String s="abc";	Ex:- StringBuffer s=new StringBuffer ("abc");														
	b	Define a class circle having data members pi and radius. Initialize and display values of data members also calculate area of circle and display it.													
	Ans	class abc {	correct Program with correct logic 4 Mark												



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<pre>float pi,radius; abc(float p, float r) { pi=p; radius=r; } void area() { float ar=pi*radius*radius; System.out.println("Area="+ar); } void display() { System.out.println("Pi="+pi); System.out.println("Radius="+radius); } } class area { public static void main(String args[]) { abc a=new abc(3.14f,5.0f); a.display();</pre>	
--	--	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		<pre>a.area(); } }</pre>	
	c	Define exception. State built-in exceptions.	4M
	Ans	<p>An exception is a problem that arises during the execution of a program.</p> <p>Java exception handling is used to handle error conditions in a program systematically by taking the necessary action</p> <p>Built-in exceptions:</p> <ul style="list-style-type: none">• Arithmetic exception: Arithmetic error such as division by zero.• ArrayIndexOutOfBoundsException: Array index is out of bound• ClassNotFoundException• FileNotFoundException: Caused by an attempt to access a nonexistent file.• IO Exception: Caused by general I/O failures, such as inability to read from a file.• NullPointerException: Caused by referencing a null object.• NumberFormatException: Caused when a conversion between strings and number fails.• StringIndexOutOfBoundsException: Caused when a program attempts to access a nonexistent character position in a string.• OutOfMemoryException: Caused when there's not enough memory to allocate a new object.• SecurityException: Caused when an applet tries to perform an action not allowed by the browser's security setting.• StackOverflowException: Caused when the system runs out of stack space.	Definition 2 Marks, List: 2 Marks
	d	Write syntax and example of :	4M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		1) drawRect() 2)drawOval()	
	Ans	1)drawRect() : drawRect () method display an outlined rectangle. Syntax: void drawRect(int top,int left, int width,int height) The upper-left corner of the Rectangle is at top and left. The dimension of the Rectangle is specified by width and height. Example: g.drawRect(10,10,60,50); 2) drawOval(): Drawing Ellipses and circles: To draw an Ellipses or circles used drawOval () method can be used. Syntax: void drawOval(int top, int left, int width, int height) The ellipse is drawn within a bounding rectangle whose upper-left corner is specified by top and left and whose width and height are specified by width and height to draw a circle or filled circle, specify the same width and height the following program draws several ellipses and circle. Example: g.drawOval(10,10,50,50);	drawRect: 2Marks, drawOval: 2 Marks
3.		Attempt any Three of the following:	
	a	Explain the following classes. i)Byte stream class ii)Character Stream Class	4M
	Ans	i)Byte stream class: 1) InputStream and OutputStream are designed for byte streams 2) Use the byte stream classes when working with bytes or other binary objects. 3) Input Stream is an abstract class that defines Java's model of streaming byte input	2M for any two points

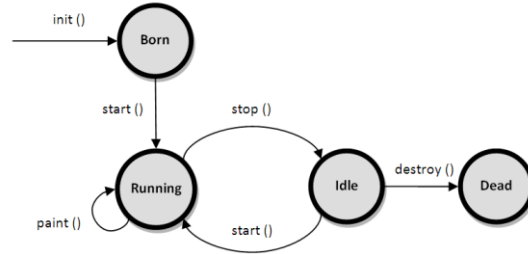


MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<p>4)The Input stream class defines methods for performing input function such as reading bytes, closing streams, Marking position in stream.</p> <p>5) Output Stream is an abstract class that defines streaming byte output.</p> <p>6) The output stream class defines methods for performing output function such as writing bytes, closing streams</p> <p>ii)Character Stream Class:</p> <ol style="list-style-type: none">1. Reader and Writer are designed for character streams.2. Use character stream classes when working with characters or strings.3. Writer stream classes are designed to write characters.4. Reader stream classes are designed to read characters. <p>5)The two subclasses used for handling characters in file are FileReader (for reading characters) and FileWriter (for writing characters).</p>	
b	Explain life cycle of Applet.	4M
Ans	<p>When an applet begins, the AWT calls the following methods, in this sequence:</p> <ol style="list-style-type: none">1. init()2. start()3. paint() <p>When an applet is terminated, the following sequence of method calls takes place:</p> <ol style="list-style-type: none">4. stop()5. destroy()	1M for diagram ,3M for explanation



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)



init ():The **init()** method is the first method to be called. This is where you should initialize Variables. This method is called only once during the run time of your applet.

start():The **start()** method is called after **init()**.It is also called to restart an applet after it has Been stopped. Whereas **init()** is called once—the first time an applet is loaded—**start()** is called each time an applet’s HTML document is displayed onscreen.

Paint (): The **paint ()** method is called each time your applet’s output must be redrawn. Paint () is also called when the applet begins execution. Whatever the cause, whenever the applet must redraw its output, paint() is called. The paint () method has one parameter of type Graphics.

Stop (): When stop () is called, the applet is probably running. You should use stop () to suspend threads that don’t need to run when the applet is not visible.

destroy(): The destroy () method is called when the environment determines that your applet needs to be removed completely from memory.

c	Differentiate between class and interfaces.	4M								
Ans	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Class</th> <th style="width: 50%; text-align: center;">Interface</th> </tr> </thead> <tbody> <tr> <td>1)doesn’t Supports multiple inheritance</td> <td>1) Supports multiple inheritance</td> </tr> <tr> <td>2)”extend ” keyword is used to inherit</td> <td>2)”implements ” keyword is used to inherit</td> </tr> <tr> <td>3) class contain method body</td> <td>3) interface contains abstract method(method without body)</td> </tr> </tbody> </table>	Class	Interface	1)doesn’t Supports multiple inheritance	1) Supports multiple inheritance	2)”extend ” keyword is used to inherit	2)”implements ” keyword is used to inherit	3) class contain method body	3) interface contains abstract method(method without body)	1M for each point
Class	Interface									
1)doesn’t Supports multiple inheritance	1) Supports multiple inheritance									
2)”extend ” keyword is used to inherit	2)”implements ” keyword is used to inherit									
3) class contain method body	3) interface contains abstract method(method without body)									



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	4)contains any type of variable	4)contains only final variable	
	5)can have constructor	5)cannot have constructor	
	6)can have main() method	6)cannot have main() method	
	7)syntax Class classname { Variable declaration, Method declaration }	7)syntax Inteface Innterfacename { Final Variable declaration, abstract Method declaration }	
d	Define type casting. Explain its types with syntax and example.		4M
Ans	<p>1. The process of converting one data type to another is called casting or type casting.</p> <p>2. If the two types are compatible, then java will perform the conversion automatically.</p> <p>3. It is possible to assign an int value to long variable.</p> <p>4. However, if the two types of variables are not compatible, the type conversions are not implicitly allowed, hence the need for type casting.</p> <p>There are two types of conversion:</p> <p>1.Implicit type-casting:</p> <p>2.Explicit type-casting:</p> <p>1. Implicit type-casting:</p> <p>Implicit type-casting performed by the <i>compiler automatically</i>; if there will be no loss of precision.</p> <p>Example:</p> <pre>int i = 3; double f; f = i;</pre>		1M for definition,3M for types explanation

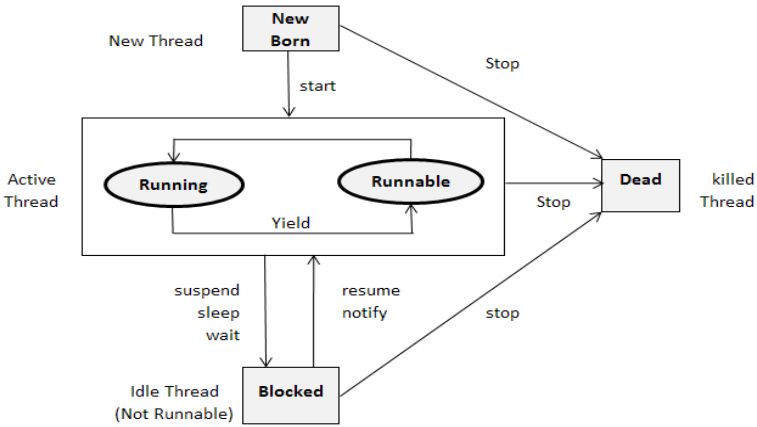


MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		<p>output: f = 3.0</p> <p>Widening Conversion:</p> <p>The rule is to promote the smaller type to bigger type to prevent loss of precision, known as Widening Conversion.</p> <p>2. Explicit type-casting:</p> <ul style="list-style-type: none">• Explicit type-casting performed via a type-casting operator in the prefix form of (<i>new-type</i>) operand.• Type-casting forces an explicit conversion of type of a value. Type casting is an operation which takes one operand, operates on it and returns an equivalent value in the specified type. <p>Syntax:</p> <p>newValue = (typecast)value;</p> <p>Example:</p> <p>double f = 3.5;</p> <p>int i; i = (int)f; // it cast double value 3.5 to int 3.</p> <p>Narrowing Casting: Explicit type cast is requires to Narrowing conversion to inform the compiler that you are aware of the possible loss of precision.</p>	
4.		Attempt any Three of the following:	
	a	Explain life cycle of thread.	4M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

<p>Ans</p>	 <p>The diagram illustrates the Thread Life Cycle with five states: New Born, Running, Runnable, Blocked, and Dead. New Born (New Thread) transitions to Running (Active Thread) via start and to Dead (killed Thread) via Stop. Running and Runnable are grouped as Active Thread. Running can transition to Runnable via Yield and back to Running. Runnable can transition to Running via Yield. Running can transition to Blocked (Idle Thread, Not Runnable) via suspend, sleep, or wait. Blocked can transition to Runnable via resume or notify. Blocked can transition to Dead via stop. Runnable can transition to Dead via Stop.</p> <p>Thread Life Cycle Thread has five different states throughout its life.</p> <ol style="list-style-type: none">1. Newborn State2. Runnable State3. Running State4. Blocked State5. Dead State <p>Thread should be in any one state of above and it can be move from one state to another by different methods and ways.</p> <p>Newborn state: When a thread object is created it is said to be in a new born state. When the thread is in a new born state it is not scheduled running from this state it can be scheduled for running by start() or killed by stop(). If put in a queue it moves to runnable state.</p> <p>Runnable State: It means that thread is ready for execution and is waiting for the availability of the processor i.e. the thread has joined the queue and is waiting for execution. If all threads have equal priority, then they are given time slots for execution in round robin fashion. The thread that relinquishes control joins the queue at the end and again waits for its turn. A thread can relinquish the control to another before its turn comes by yield().</p>	<p>2M for diagram, 2M for explanation</p>
-------------------	---	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<p>Running State: It means that the processor has given its time to the thread for execution. The thread runs until it relinquishes control on its own or it is pre-empted by a higher priority thread.</p> <p>Blocked state: A thread can be temporarily suspended or blocked from entering into the runnable and running state by using either of the following thread method.</p> <ol style="list-style-type: none">1) suspend() : Thread can be suspended by this method. It can be rescheduled by resume().2) wait(): If a thread requires to wait until some event occurs, it can be done using wait method and can be scheduled to run again by notify().3) sleep(): We can put a thread to sleep for a specified time period using sleep(time) where time is in ms. It re-enters the runnable state as soon as period has elapsed /over <p>Dead State: Whenever we want to stop a thread form running further we can call its stop().The statement causes the thread to move to a dead state. A thread will also move to dead state automatically when it reaches to end of the method. The stop method may be used when the premature death is required.</p>	
b	Describe final variable and final method.	4M
Ans	<p>Final method: making a method final ensures that the functionality defined in this method will never be altered in any way, ie a final method cannot be overridden.</p> <p>Syntax:</p> <pre>final void findAverage() { //implementation }</pre> <p>Example of declaring a final method:</p> <pre>class A {</pre>	2M for definition,2M for example



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<pre>final void show() { System.out.println("in show of A"); } } class B extends A { void show() // can not override because it is declared with final { System.out.println("in show of B"); }} Final variable: the value of a final variable cannot be changed. Final variable behaves like class variables and they do not take any space on individual objects of the class. Example of declaring final variable: final int size = 100;</pre>									
c	Explain any two logical operator in java with example.	4M								
Ans	<p>Logical Operators: Logical operators are used when we want to form compound conditions by combining two or more relations. Java has three logical operators as shown in table:</p> <table border="1"><thead><tr><th>Operator</th><th>Meaning</th></tr></thead><tbody><tr><td>&&</td><td>Logical AND</td></tr><tr><td> </td><td>Logical OR</td></tr><tr><td>!</td><td>Logical NOT</td></tr></tbody></table> <p>Program demonstrating logical Operators</p> <pre>public class Test</pre>	Operator	Meaning	&&	Logical AND		Logical OR	!	Logical NOT	2M for each operator with eg.
Operator	Meaning									
&&	Logical AND									
	Logical OR									
!	Logical NOT									



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
 (Autonomous)
 (ISO/IEC - 27001 - 2013 Certified)

		<pre> { public static void main(String args[]) { boolean a = true; boolean b = false; 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) = true </pre>					
	d	Differentiate between array and vector.	4M				
	Ans	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;">Array</th> <th style="width: 50%; text-align: left;">Vector</th> </tr> </thead> <tbody> <tr> <td>1) An array is a structure that holds multiple values of the same type.</td> <td>1)The Vector is similar to array holds multiple objects and like an array; it contains components that can be accessed using an integer index.</td> </tr> </tbody> </table>	Array	Vector	1) An array is a structure that holds multiple values of the same type.	1)The Vector is similar to array holds multiple objects and like an array; it contains components that can be accessed using an integer index.	any four points 1m for each point
Array	Vector						
1) An array is a structure that holds multiple values of the same type.	1)The Vector is similar to array holds multiple objects and like an array; it contains components that can be accessed using an integer index.						



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<p>2) An array is a homogeneous data type where it can hold only objects of one data type.</p> <p>3) After creation, an array is a fixed-length structure.</p> <p>4) Array can store primitive type data element.</p> <p>5) Declaration of an array : int arr[] = new int [10];</p> <p>6) Array is the static memory allocation.</p>	<p>2) Vectors are heterogeneous. You can have objects of different data types inside a Vector.</p> <p>3) The size of a Vector can grow or shrink as needed to accommodate adding and removing items after the Vector has been created.</p> <p>4) Vector are store non-primitive type data element.</p> <p>5) Declaration of Vector: Vector list = new Vector(3);</p> <p>6) Vector is the dynamic memory allocation.</p>	
e	List any four methods of string class and state the use of each.		4M
Ans	<p>The java.lang.String class provides a lot of methods to work on string. By the help of these methods,</p> <p>We can perform operations on string such as trimming, concatenating, converting, comparing, replacing strings etc.</p> <p>1) to Lowercase (): Converts all of the characters in this String to lower case.</p> <p>Syntax: s1.toLowerCase() Example: String s="Sachin"; System.out.println(s.toLowerCase()); Output: sachin</p> <p>2)to Uppercase():Converts all of the characters in this String to upper case</p>		any four methods of string class can be considered



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		<p>Syntax: <code>s1.toUpperCase()</code></p> <p>Example:</p> <pre>String s="Sachin"; System.out.println(s.toUpperCase());</pre> <p>Output: SACHIN</p> <p>3) trim (): Returns a copy of the string, with leading and trailing whitespace omitted.</p> <p>Syntax: <code>s1.trim()</code></p> <p>Example:</p> <pre>String s=" Sachin "; System.out.println(s.trim());</pre> <p>Output:Sachin</p> <p>4) replace ():Returns a new string resulting from replacing all occurrences of old Char in this string with new Char.</p> <p>Syntax: <code>s1.replace('x','y')</code></p> <p>Example:</p> <pre>String s1="Java is a programming language. Java is a platform." String s2=s1.replace("Java","Kava");//replaces all occurrences of "Java" to "Kava" System.out.println(s2);</pre> <p>Output: Kava is a programming language. Kava is a platform.</p>	
5.		Attempt any Three of the following:	12-Total Marks
	a	Write a program to create a vector with five elements as (5, 15, 25, 35, 45). Insert new element at 2nd position. Remove 1st and 4th element from vector.	6M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

Ans	<pre>import java.util.*; class VectorDemo { public static void main(String[] args) { Vector v = new Vector(); v.addElement(new Integer(5)); v.addElement(new Integer(15)); v.addElement(new Integer(25)); v.addElement(new Integer(35)); v.addElement(new Integer(45)); System.out.println("Original array elements are "); for(int i=0;i<v.size();i++) { System.out.println(v.elementAt(i)); } v.insertElementAt(new Integer(20),1); // insert new element at 2nd position v.removeElementAt(0); //remove first element v.removeElementAt(3); //remove fourth element System.out.println("Array elements after insert and remove operation "); for(int i=0;i<v.size();i++) { System.out.println(v.elementAt(i)); } } }</pre>	<p><i>(Vector creation with elements – 2 M,</i></p> <p><i>Insert new element – 2M,</i></p> <p><i>Remove elements 2 M,</i></p> <p>(Any other logic can be considered)</p>
b	Define package. How to create user defined package? Explain with example.	6M
Ans	Java provides a mechanism for partitioning the class namespace into more manageable parts. This mechanism is the package. The package is both naming and visibility controlled mechanism. Package can be created by including package as the first statement in java source code. Any classes declared within that file will belong to the specified package. Package defines a namespace in	(Definition of package - 1M,



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<p>which classes are stored.</p> <p>The syntax for defining a package is: package <i>pkg</i>; Here, <i>pkg</i> is the name of the package eg : package mypack;</p> <p>Packages are mirrored by directories. Java uses file system directories to store packages. The class files of any classes which are declared in a package must be stored in a directory which has same name as package name. The directory must match with the package name exactly. A hierarchy can be created by separating package name and sub package name by a period(.) as <i>pkg1.pkg2.pkg3</i>; which requires a directory structure as <i>pkg1\pkg2\pkg3</i>.</p> <p>Syntax: To access package In a Java source file, import statements occur immediately following the package statement (if it exists) and before any class definitions.</p> <p>Syntax: import <i>pkg1</i>[.<i>pkg2</i>].(<i>classname</i> *);</p> <p>Example: package package1; public class Box { int l= 5; int b = 7; int h = 8; public void display() { System.out.println("Volume is:"+(l*b*h)); } }</p> <p>Source file: import package1.Box; class volume {</p>	<p>Package creation - 2M</p> <p>Example - 3M</p> <p>(Note Any other example can be considered)</p>
--	---	---



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<pre>public static void main(String args[]) { Box b=new Box(); b.display(); } }</pre>	
c	Write a program to create two threads one thread will print even no. between 1 to 50 and other will print odd number between 1 to 50.	6M
Ans	<pre>import java.lang.*; class Even extends Thread { public void run() { try { for(int i=2;i<=50;i=i+2) { System.out.println("\t Even thread :"+i); sleep(500); } } catch(InterruptedException e) {System.out.println("even thread interrupted"); } } } class Odd extends Thread { public void run() { try { for(int i=1;i<50;i=i+2) { System.out.println("\t Odd thread :"+i); sleep(500); } } } }</pre>	Creation of two threads 4M Creating main to create and start objects of 2 threads: 2M (Any other logic can be considered)



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		<pre> } } catch(InterruptedException e) {System.out.println("odd thread interrupted"); } } } class EvenOdd { public static void main(String args[]) { new Even().start(); new Odd().start(); } } </pre>	
6.		Attempt any Three of the following:	12 M
	a	Explain how to pass parameter to an applet ? Write an applet to accept username in the form of parameter and print “Hello <username>”.	6M
	Ans	<p>Passing Parameters to Applet</p> <ul style="list-style-type: none"> • User defined parameters can be supplied to an applet using <PARAM.....> tags. • PARAM tag names a parameter the Java applet needs to run, and provides a value for that parameter. • PARAM tag can be used to allow the page designer to specify different colors, fonts, URLs or other data to be used by the applet. <p>To set up and handle parameters, two things must be done.</p> <p>1. Include appropriate <PARAM..>tags in the HTML document.</p> <p>The Applet tag in HTML document allows passing the arguments using param tag. The syntax of <PARAM...> tag</p> <pre> <Applet code="AppletDemo" height=300 width=300> <PARAM NAME = name1 VALUE = value1> </Applet> NAME:attribute name VALUE: value of attribute named by corresponding PARAM NAME. </pre>	<p>(Explanation for parameter passing - 3M,</p> <p>Correct Program – 3M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<p>2. Provide code in the applet to parse these parameters. The Applet access their attributes using the getParameter method.</p> <p>The syntax is : String getParameter(String name);</p> <p>Program</p> <pre>import java.awt.*; import java.applet.*; public class hellouser extends Applet { String str; public void init() { str = getParameter("username"); str = "Hello "+ str; } public void paint(Graphics g) { g.drawString(str,10,100); } } <HTML> <Applet code = hellouser.class width = 400 height = 400> <PARAM NAME = "username" VALUE = abc> </Applet> </HTML></pre> <p>(OR)</p> <pre>import java.awt.*; import java.applet.*; /*<Applet code = hellouser.class width = 400 height = 400> <PARAM NAME = "username" VALUE = abc> </Applet>*/ public class hellouser extends Applet { String str; public void init() { str = getParameter("username"); str = "Hello "+ str; } }</pre>	
--	---	--

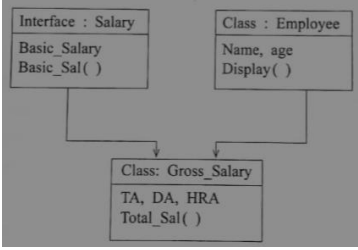


MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<pre>public void paint(Graphics g) { g.drawString(str,10,100); } }</pre>	
b	Write a program to perform following task (i) Create a text file and store data in it. (ii) Count number of lines and words in that file.	6M
Ans	<pre>import java.util.*; import java.io.*; class Model6B { public static void main(String[] args) throws Exception { int lineCount=0, wordCount=0; String line = ""; BufferedReader br1 = new BufferedReader(new InputStreamReader(System.in)); FileWriter fw = new FileWriter("Sample.txt"); //create text file for writing System.out.println("Enter data to be inserted in file: "); String fileData = br1.readLine(); fw.write(fileData); fw.close(); BufferedReader br = new BufferedReader(new FileReader("Sample.txt")); while ((line = br.readLine()) != null) { lineCount++; // no of lines count String[] words = line.split(" "); wordCount = wordCount + words.length; // no of words count } System.out.println("Number of lines is : " + lineCount); System.out.println("Number of words is : " + wordCount); } }</pre>	Create file and store data : 3M, Get lines and word count : 3M) (Any other logic can be considered)



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

		}	
	c	Implement the following inheritance 	6M
	Ans	<pre>interface Salary { double Basic Salary=10000.0; void Basic Sal(); } class Employee { String Name; int age; Employee(String n, int b) { Name=n; age=b; } void Display() { System.out.println("Name of Employee :"+Name); System.out.println("Age of Employee :"+age); } } class Gross_Salary extends Employee implements Salary { double HRA,TA,DA; Gross_Salary(String n, int b, double h,double t,double d) { super(n,b); HRA=h;</pre>	(Interface: 1M, Employee class: 2M, Gross_Salary class: 3M)



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

	<pre>TA=t; DA=d; } public void Basic_Sal() { System.out.println("Basic Salary :"+Basic_Salary); } void Total_Sal() { Display(); Basic_Sal(); double Total_Sal=Basic_Salary + TA + DA + HRA; System.out.println("Total Salary :"+Total_Sal); } } class EmpDetails { public static void main(String args[]) { Gross_Salary s=new Gross_Salary("Sachin",20,1000,2000,7000); s.Total_Sal(); } }</pre>	<p>(Any other logic considered)</p>
--	---	--