BHARATI VIDYAPEETH DEEMED UNIVERSITY
COLLEGE OF ENGINEERING, PUNE - 43

DEPARTMENT OF COMPUTER ENGINEERING

Laboratory Manual

Programming Lab -I

B.Tech. Computer Sem III

DEPARTMENT OF COMUTER ENGINEERING

VISION OF THE INSTITUTE

To be World Class Institute for Social Transformation Through Dynamic Education.

MISSION OF THE INSTITUTE

A. To provide quality technical education with advanced equipment, qualified faculty members, infrastructure to meet needs of profession and society.

B. To provide an environment conducive to innovation, creativity, research, and entrepreneurial leadership.

C. To practice and promote professional ethics, transparency and accountability for social community, economic and environmental conditions.

VISION OF THE DEPARTMENT

To pursue and excel in the Endeavour for creating globally recognized Computer Engineers through Quality education.

MISSION OF THE DEPARTMENT

- To impart engineering knowledge and skills confirming to a dynamic curriculum.
- To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth.
- To produce qualified graduates exhibiting societal and ethical responsibilities in working environment.

PROGRAM EDUCATIONAL OBJECTIVES

1. Demonstrate technical and professional competencies by applying engineering fundamentals, computing principles and technologies.
2. Learn, Practice, and grow as skilled professionals/ entrepreneur/researchers adapting to the evolving computing landscape.
3. Demonstrate professional attitude, ethics, understanding of social context and interpersonal skills leading to a successful career.

PROGRAM SPECIFIC OUTCOMES

1. To apply fundamental knowledge and technical skills towards solving Engineering problems.
2. To employ expertise and ethical practise through continuing intellectual growth and adapting to the working environment.

PROGRAM OUTCOMES

1. To apply knowledge of computing and mathematics appropriate to the domain.
2. To logically define, analyze and solve real world problems.
3. To apply design principles in developing hardware/software systems of varying complexity that meet the specified needs.
4. To interpret and analyze data for providing solutions to complex engineering problems.
5. To use and practice engineering and IT tools for professional growth.
6. To understand and respond to legal and ethical issues involving the use of technology for societal benefits.
7. To develop societal relevant projects using available resources.
8. To exhibit professional and ethical responsibilities.
9. To work effectively as an individual and a team member within the professional environment.
10. To prepare and present technical documents using effective communication skills.
11. To demonstrate effective leadership skills throughout the project management life cycle.
12. To understand the significance of lifelong learning for professional development.

GENRAL INSTUCTIONS:

- Equipment in the lab is meant for the use of students. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care.
- Students are required to carry their reference materials, files and records with completed assignment while entering the lab.
- Students are supposed to occupy the systems allotted to them and are not supposed to talk or make noise in the lab.
- All the students should perform the given assignment individually.
- Lab can be used in free time/lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.
- All the Students are instructed to carry their identity cards when entering the lab.
- Lab files need to be submitted on or before date of submission.
- Students are not supposed to use pen drives, compact drives or any other storage devices in the lab.
- For Laboratory related updates and assignments students should refer to the notice board in the Lab.

Programme: B. Tech. (Computer) – SEM III – 2014 Course

Sr. no.	Subject	Teaching Scheme (Hrs/Week)			Examination Scheme (Marks)							Credits		
		L	P/D	T	End Semester Examination	Continuous Assessment			TW & Practical	TW & Oral	Total	Theory	TW	Total
						Unit Test	Attendance	Assignments						
1	Fundamentals of Data Communication	3	--	1	60	20	10	10	--	--	100	4	--	4
2	Principles of Data Structures	3	2	--	60	20	10	10	50	--	150	3	1	4
3	Digital Techniques and Logic Design	3	2	--	60	20	10	10	--	50	150	3	1	4
4	Discrete Mathematics and Graph Theory	3	--	--	60	20	10	10	--	--	100	3	--	3
5	Engineering Economics and Management	3	2	--	60	20	10	10	50		150	3	1	4
6	Professional Skill Development- III	4	--	--	100	--	--	-	--	--	100	4	--	4
7	Programming Lab-I	--	4	--	--	--	--	--	50	--	50	--	2	2
	TOTAL	**19**	**10**	**1**	**400**	**100**	**50**	**50**	**150**	**50**	**800**	**20**	**05**	**25**

07: PROGRAMMING LABORATORY I

TEACHING SCHEME:	**EXAMINATION SCHEME:**	**CREDITS ALLOTTED:**
Practical: 04 Hours / Week	Term Work (Practical): 50 Marks	02 Credits

Course Pre-requisites:

The Students should have

1. Basics concepts of Object Oriented Programming

2. Basic mathematical ability

Course Objectives:

1. At the completion of this course, student is able to read, write, and debug java programs using good programming style.

Course Outcomes:

1. Write java programs using JDK & Solve arithmetic expressions using java program.

2. Gain knowledge & apply concepts of class fundamentals in various programming assignments.

3. Differentiate between String & String Buffer Class. And use different functions of these classes in various programming assignments.

4. Understand and implement the concept of interfaces and packages.

5. Learn the concept of Exception handling and Apply it.

6. Apply the functions of AWT classes in various programming assignments.

UNIT - I **Java Evolution**: Difference between Java , C, C++, Features of java, The java runtime environment (JDK, JVM, Command Line Arguments),Sample java program, Java statements and program structure, Fundamental programming **(06 Hours)**

constructs in java**: (**Constants, Variables, keywords, Data Types, Operators, Expressions and control structures)

UNIT - II

Classes and methods: Specification of a class, Introduction to Methods, Access specifiers, Constructors, Method overloading, this keyword, finalizer method, recursion, Introducing Final keyword , Concept of array, Introducing Nested and Inner Classes, Inheritance , Using Super, Method overriding, Dynamic method Dispatch, Abstract class concept.

(06 Hours)

UNIT - III

String Handling:

(06 Hours)

The String Constructors, String Operations, CharacterExtraction, StringComparison, Modifying a String, String Buffer.

The Collections Framework:

More Utility Classes, Networking, the Applet Class, Event Handling, Collection Interface, List Interface, Set interface, Map Interface , Enumeration Introduction to Swing.

UNIT - IV

Package and Interfaces:

(06 Hours)

Introduction to package: Types of packages, User define packages, Use of package keyword, Importing packages.

Interfaces: Define and implement interface, use of interfaces to support multiple inheritance, variables in interfaces, interfaces can be extended.

UNIT - V

Exception Handling and Multithreaded programming:

(06 Hours)

Exception handling: Introduction to exception handling, predefined and user defined exceptions.

Use of try, catch, throw, throws and finally keywords.

Introduction to threads, life cycle of a thread, thread states, thread properties, methods in Threads and Runnable, setting priority of threads, synchronization and inter thread communication

UNIT - VI	**Introducing the AWT, Using AWT controls, Images:** Introduction to AWT, events, listeners, event handling methods, a small application to demonstrate use of controls – label, button, check box, text, radio button, Dialog Box, scroll bar, choice controls ,List, Menu bars and Menus layout. Image Fundamentals image Observer, Double Buffering, and ImageProducer. AWT classes, Frame windows, Creating a Frame Window in an Applet, Working with graphics, Working with color, Setting the Paint Mode, Working with Fonts, Control Fundamentals,	**(06 Hours)**

Term Work:

1. Introduction to Java.

2. Write a program to implement Class and Inheritance Concept.

3. Write a program to differentiate between method overloading and method overriding.

4. Write a program to understand the use of String class and string buffer class

5. Write a program to implement Applet.

6. Write a program to implement the concept of Package.

7. Write a program to implement concept of Exception Handling.

8. Write a program to implement the concept of Multithreaded Programming

9. Write a program to implement Frame and different graphics objects.

10. Write a program to use different controls of AWT classes.

Text Books:

1. E. Balagurusamy, Programming with Java, 3 e, McGraw-Hill Companies.
2. JAVA 7 Programming, Black Book ,Kogent Learning Solutions Inc.

Reference Books:

1. The complete reference Java 2 Third Edition,TMH publication by Patrick Naughton, Herbert Schildt.

2. Ken Arnold, James Gosling, David Holmes, “The Java Programming Language”, 3e, Sun Microsystems.

COURSE NAME: Programming Lab -I

CONTACT HOURS: 04 Hrs./Week

WEEKLY PLAN:

Week No.	Practical/Assignment Name	Problem Definition
1	Introduction to Java ,Java Evolution	Write a study assignment on Java Programming.
2	Study of Class and Object concept	Write a program to implement concept of class and object.
3	Implement concept of different types of Inheritance.	Write a program to implement concept of static member and inheritance.
4	Study and implement Keywords as final, super, this, static	Demonstrate the use of following keyword in Java. final b) super c) this d) static
5	Implement the difference between method overloading and method overriding	Write a program to differentiate between method overloading and method overriding.
6	Understand the use of String, StringBuffer, StringBuilder class.	Write a program to implement methods of String, StringBuffer and StringBuilder class.
7	Write a program to implement Applet.	Write a program for interactive applet to a) Find addition of two numbers. b) Display image on applet. c)Draw three rectangle and fill rectangle using RGB color.
8	Introduction to package: Types of packages, User define packages, Use of package keyword	Write a program to implement concept of interface and package.
9	Concept of Exception Handling	Write a program to implement the concept of exception handling.
10	Study the concept of multithreading	Write a program to implement multithreading in Java.
11	Study of different AWT controls	Write a program to implement different graphics object and controls of AWT objects/classes.

EXAMINATION SCHEME

Practical Exam: 25 Marks

Term Work: 25 Marks

Total (TW&PR): 50 Marks

Minimum Marks required: 20 Marks

PROCEDURE OF EVALUATION

Each practical/assignment shall be assessed continuously on the scale of 25 marks. The distribution of marks as follows.

Sr. No	Evaluation Criteria	Marks for each Criteria	Rubrics
1	**Timely Submission**	**07**	➢ Punctuality reflects the work ethics. Students should reflect that work ethics by completing the lab assignments and reports in a timely manner without being reminded or warned.
2	**Presentation**	**06**	➢ Student are expected to write the technical document (lab report) in their own words. The presentation of the contents in the lab report should be complete, unambiguous, clear, understandable. The report should document approach/algorithm/design and code with proper explanation.
3	**Understanding**	**12**	➢ Correctness and Robustness of the code is expected. The Learners should have an in-depth knowledge of the practical assignment performed. The learner should be able to explain methodology used for designing and developing the program/solution. He/she should clearly understand the purpose of the assignment and its outcome.

LABORATORY USAGE

Total No. of machines required :20 m/c

Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.

Software Requirement

Turbo C complier in Windows XP or Linux Operating System.

Students use machine of above configuration for executing the lab experiments, document the results and to prepare the technical documents for making the lab reports.

OBJECTIVES:-

1. To enable the students to understand basic concepts of operating system.
2. To brief the students about various design aspects of operating system functionality.

PRACTICAL PRE-REQUISITE:-

Basic knowledge of System Programming and Concepts of System Software, Application Programs, Microprocessor Basics.
Knowledge of C programming and basic UNIX commands.

COURSE OUTCOMES

1. Write java programs using JDK & Solve arithmetic expressions using java program.
2. Gain knowledge & apply concepts of class fundamentals in various programming assignments.
3. Differentiate between String & String Buffer Class. And use different functions of these classes in various programming assignments.
4. Understand and implement the concept of interfaces and packages.
5. Learn the concept of Exception handling and Apply it.
6. Apply the functions of AWT classes in various programming assignments.

HOW OUTCOMES ARE ASSESSED?

Outcome	Assignment Number	Level	Proficiency evaluated by

Write java programs using JDK & Solve arithmetic expressions using java program.	**1,2,8**	**H,H,H**	Problem definition ,Examine theoretical aspects and reporting results
Gain knowledge & apply concepts of class fundamentals in various programming assignments.	**1,2,3**	**M,H, M**	Performing Practical and reporting results
Differentiate between String & String Buffer Class. And use different functions of these classes in various programming assignments.	**1,6**	**M,M**	Performing Practical and reporting results
Understand and implement the concept of interfaces and packages.	**1,2,9**	**M,H, H**	Performing Practical and reporting results
Learn the concept of Exception handling and Apply it.	**1,2,10**	**M,H, H**	Problem definition ,Examine theoretical aspects ,Perform Practical and reporting results
Apply the functions of AWT classes in various programming assignments.	**1,2,12**	**M,M, H**	Performing experiments and reporting results

CONTRIBUTION TO PROGRAM OUTCOMES

	Program Outcomes												**PSOs**	
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PS O 2**
Course outcomes	**3**	**3**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	1	
	3	**3**	**3**	**3**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**3**		3
	3	**3**	**2**	**3**	**1**	**1**	**3**	**1**	**3**	**1**	**2**	**2**		
	3	**3**	**2**	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**1**	**3**	2	3
	3	**3**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**3**		
	3	**3**	**1**	**3**	**1**	**1**	**3**	**1**	**3**	**1**	**3**	**3**	1	3

DESIGN EXPERIENCE GAINED

Students will gain practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in Linux environment.

LEARNING EXPERIENCE GAINED

Upon the completion of Operating Systems practical course, the student will be able to:

1. Understand and implement basic services and functionalities of the operating system using system calls.
2. Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
3. Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.
4. Implement memory management schemes and page replacement schemes.
5. Simulate file allocation and organization techniques.

LIST OF PRACTICAL ASSIGNMENTS:

No	List of Assignment
1.	Write a study assignment on Java Programming.
2.	Write a program to implement concept of class and object.
3.	Write a program to implement concept of static member and inheritance.
4.	Demonstrate the use of following keyword in Java. final b) super c) this d) static
5.	Write a program to differentiate between method overloading and method overriding
6.	Write a program to implement methods of String, StringBuffer and StringBuilder class.
7.	Write a program to iterate through all elements in a array list.
8.	Write a program for interactive applet to a) Find addition of two numbers. b) Display image on applet. c)Draw three rectangle and fill rectangle using RGB color.
9.	Write a program to implement concept of interface and package.
10.	Write a program to implement the concept of exception handling.

11.	Write a program to implement multithreading in Java.
12.	Write a program to implement different graphics object and controls of AWT objects/classes.

ASSIGNMENT NO: 1

AIM: Write a study assignment on Java Programming.

OBJECTIVE: To enable the students to understand basic concepts of java.

BRIEF THEORY:
Java is a programming language and a platform.Java is a high level, robust, secured and object-oriented programming language.

Platform:
Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

History:
The history of java starts from Green Team. Java team members (also known as **Green Team**), initiated a revolutionary task to develop a language for digital devices such as set-top boxes, televisions etc. For the green team members, it was an advance concept at that time. But, it was suited for internet programming. Later, Java technology as incorporated by Netscape.Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc.

Features:
There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

Object-oriented

Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Platform Independent
A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform.
The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms.

Robust

Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.

Architecture-neutral

There is no implementation dependent features e.g. size of primitive types is fixed.In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

High-performance

Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

Conclusion: Basic concepts of Java programming along with some features are discussed here.

Questions:-

Q. What is the difference between C++ and java?

ASSIGNMENT NO: 2

AIM: Write a program to implement concept of class and object.

OBJECTIVE: To understand and implement various concepts of class and object.

BRIEF THEORY:

Object in Java

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

An object has three characteristics:
state: represents data (value) of an object.
behavior: represents the behavior (functionality) of an object such as deposit, withdraw etc.
identity: Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely.
For Example: Pen is an object. Its name is Reynolds, color is white etc. known as its state. It is used to write, so writing is its behavior.
Object is an instance of a class. Class is a template or blueprint from which objects are created. So object is the instance(result) of a class.

Class in Java

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
A class in Java can contain:
fields
methods
constructors
blocks
nested class and interface

Pseudo code:

In this example, we have created a Student class that have two data members id and name. We are creating the object of the Student class by new keyword and printing the objects value.
Here, we are creating main() method inside the class.

```
class Student{
int id;//field or data member or instance variable
String name;

public static void main(String args[]){
Student s1=new Student();//creating an object of Student
System.out.println(s1.id);//accessing member through reference variable
System.out.println(s1.name);
}
}
```

Output:

```
0
null
```

Conclusion: Implementation through concepts of class and objects is done here.

Questions:-

Q. What is the difference between an object and a class?

ASSIGNMENT NO: 3

AIM: Write a program to implement concept of static member and inheritance.

OBJECTIVE: Understanding the concepts of static member and inheritance.

BRIEF THEORY: The class level members which have static keyword in their definition are called static members.

Types of Static Members:
Java supports four types of static members
Static Variables
Static Blocks
Static Methods
Main Method (static method)

- All static members are identified and get memory location at the time of class loading by default by JVM in Method area.
- Only static variables get memory location, methods will not have separate memory location like variables.
- Static Methods are just identified and can be accessed directly without object creation.

Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object. The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.
Inheritance represents the IS-A relationship, also known as *parent-child* relationship.

Uses:
For Method Overriding (so runtime polymorphism can be achieved).
For Code Reusability.

Syntax of Java Inheritance:

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

Conclusion: Implementation through concepts like static member and inheritance is done here.

Questions:-

Q. why java main method is static?

Q. Why multiple inheritance is not supported in java?

ASSIGNMENT NO: 4

AIM: Demonstrate the use of following keyword in Java.

a) final b) super c) this d) static

OBJECTIVE: Understanding the use of various keywords.

BRIEF THEORY:

Final Keyword In Java

The final keyword in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

- variable
- method
- class

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let's first learn the basics of final keyword.

Java final variable

If you make any variable as final, you cannot change the value of final variable(It will be constant).

Pseudo code:

There is a final variable speedlimit, we are going to change the value of this variable, but It can't be changed because final variable once assigned a value can never be changed.

```
class Bike9{
final int speedlimit=90;//final variable
void run(){
speedlimit=400;
}
public static void main(String args[]){
Bike9 obj=new  Bike9();
obj.run();
}
}//end of class
```

Super Keyword In Java

The super keyword in java is a reference variable which is used to refer immediate parent class object.

Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

Usage of java super Keyword

- super can be used to refer immediate parent class instance variable.
- super can be used to invoke immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

Pseudo code:

Let's see the real use of super keyword. Here, Emp class inherits Person class so all the properties of Person will be inherited to Emp by default. To initialize all the property, we are using parent class constructor from child class. In such way, we are reusing the parent class constructor.

```
class Person{
int id;
String name;
Person(int id,String name){
this.id=id;
this.name=name;
}
}
class Emp extends Person{
float salary;
Emp(int id,String name,float salary){
super(id,name);//reusing parent constructor
this.salary=salary;
}
void display(){System.out.println(id+" "+name+" "+salary);}
}
class TestSuper5{
public static void main(String[] args){
Emp  e1=new  Emp(1,"ankit",45000f);
e1.display();
}}
```

Output:

```
1 ankit 45000
```

This Keyword In Java

There can be a lot of usage of java this keyword. In java, this is a reference variable that refers to the current object.

Usage of java this keyword

Here is given the 6 usage of java this keyword.

- this can be used to refer current class instance variable.
- this can be used to invoke current class method (implicitly)
- this() can be used to invoke current class constructor.

- this can be passed as an argument in the method call.
- this can be passed as argument in the constructor call.
- this can be used to return the current class instance from the method.

Pseudo code:

```
class Student{
int rollno;
String name;
float fee;
Student(int rollno,String name,float fee){
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
void display(){System.out.println(rollno+" "+name+" "+fee);}
}

class TestThis2{
public static void main(String args[]){
Student s1=new Student(111,"ankit",5000f);
Student s2=new Student(112,"sumit",6000f);
s1.display();
s2.display();
}}
```

Output:

```
111 ankit 5000
112 sumit 6000
```

Java Static Keyword

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

The static can be:

- variable (also known as class variable)
- method (also known as class method)
- block
- nested class

Java static variable

If you declare any variable as static, it is known static variable.
The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees,college name of students etc.
The static variable gets memory only once in class area at the time of class loading.
Pseudo code:

```
//Program of static variable

class Student8{
 int rollno;
 String name;
 static String college ="ITS";

 Student8(int r,String n){
 rollno = r;
 name = n;
 }
 void display (){System.out.println(rollno+" "+name+" "+college);}

 public static void main(String args[]){
 Student8 s1 = new Student8(111,"Karan");
 Student8 s2 = new Student8(222,"Aryan");

 s1.display();
 s2.display();
 }
 }
```

Output:

```
111 Karan ITS
222 Aryan ITS
```

Conclusion: The usage of various keywords in Java is done here.

Questions:-
Q. Is final method inherited?
Q. Can super be used to invoke parent class method?
Q. Why java main method is static?

ASSIGNMENT NO: 5

AIM: Write a program to differentiate between method overloading and method overriding.

OBJECTIVE: Understanding the basic differences between method overloading and method overriding.

BRIEF THEORY:

Method Overloading in Java

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

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

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

So, we perform method overloading to figure out the program quickly.

Advantage of method overloading

Method overloading increases the readability of the program.

Pseudo code:

```
class Adder{
static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}
class TestOverloading1{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(11,11,11));
}}
```

Output:

```
22
33
```

Method Overriding in Java

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in java**.**

In other words, If subclass provides the specific implementation of the method that has been provided by one of its parent class, it is known as method overriding.

Usage of Java Method Overriding

- Method overriding is used to provide specific implementation of a method that is already provided by its super class.
- Method overriding is used for runtime polymorphism

Pseudo code:

```
class Vehicle{
void run(){System.out.println("Vehicle is running");}
}
class Bike2 extends Vehicle{
void run(){System.out.println("Bike is running safely");}

public static void main(String args[]){
Bike2 obj = new Bike2();
obj.run();
}
```

Output:

```
Bike is running safely
```

Conclusion: Differences between method overloading and method overriding are seen here.

Questions:-

Q. Can we overload java main() method?

Q. Can we override static method?

ASSIGNMENT NO: 6

AIM: Write a program to implement methods of String, StringBuffer and StringBuilder class.

OBJECTIVE: Understanding various methods of String, StringBuffer and StringBuilder class.

BRIEF THEORY:

Java String

In java, string is basically an object that represents sequence of char values. An array of characters works same as java string. For example:

```
char[] ch={'j','a','v','a','t','p','o','i','n','t'};
String s=new String(ch);
```

is same as:

```
String s="javatpoint";
```

Java String class provides a lot of methods to perform operations on string such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc.

The java.lang.String class implements Serializable, Comparable and CharSequence interfaces.

Java StringBuffer class

Java StringBuffer class is used to create mutable (modifiable) string. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

Pseudo code:

```
class StringBufferExample{
public static void main(String args[]){
StringBuffer sb=new StringBuffer("Hello ");
sb.append("Java");//now original string is changed
System.out.println(sb);//prints Hello Java
}
}
```

Java StringBuilder class

Java StringBuilder class is used to create mutable (modifiable) string. The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized. It is available since JDK 1.5.

Pseudo code:

```
class StringBuilderExample{
public static void main(String args[]){
StringBuilder sb=new StringBuilder("Hello ");
sb.append("Java");//now original string is changed
System.out.println(sb);//prints Hello Java
}
```

}

Conclusion: Implementation methods of String, StringBuffer and StringBuilder class is done here.

Questions:-
Q. How can we create String object?
Q. What is the difference between String and StringBuffer.

ASSIGNMENT NO: 7

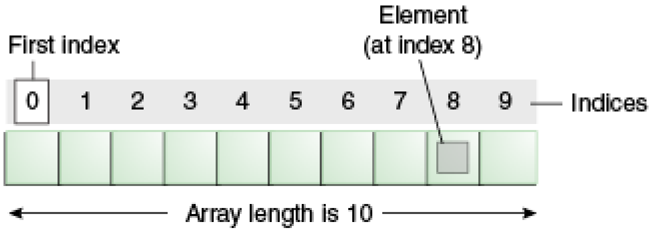
AIM: Write a program to iterate through all elements in a array list.

OBJECTIVE: Iteration of all elements in a array list.

BRIEF THEORY:

Java Array

Normally, array is a collection of similar type of elements that have contiguous memory location. Java array is an object the contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array.
Array in java is index based, first element of the array is stored at 0 index.



Advantage of Java Array

- Code Optimization: It makes the code optimized, we can retrieve or sort the data easily.
- Random access: We can get any data located at any index position.

Disadvantage of Java Array

- Size Limit: We can store only fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in java.

Conclusion: Iteration of all elements in a array list is done here.

Questions:-

Q. What is the class name of java array?

ASSIGNMENT NO: 8

AIM: Write a program for interactive applet to
a) Find addition of two numbers.
b) Display image on applet.
c) Draw three rectangle and fill rectangle using RGB color.

OBJECTIVE: Understanding the concept of interactive applet.

BRIEF THEORY:

Java Applet

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Advantage of Applet

There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plugin is required at client browser to execute applet.

Lifecycle methods for Applet:

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

java.applet.Applet class

For creating any applet java.applet.Applet class must be inherited. It provides 4 life cycle methods of applet.

- public void init(): is used to initialized the Applet. It is invoked only once.
- public void start**():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
- public void stop**():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
- public void destroy(): is used to destroy the Applet. It is invoked only once.

Conclusion: Implementation of programs on interactive applet.

Questions:-

Q. Who is responsible to manage the life cycle of an applet?

ASSIGNMENT NO: 9

AIM: Write a program to implement concept of interface and package.

OBJECTIVE: Basic understanding of interface and packages.

BRIEF THEORY:

Interface in Java

An interface in java is a blueprint of a class. It has static constants and abstract methods.The interface in java is a mechanism to achieve abstraction. There can be only abstract methods in the java interface not method body. It is used to achieve abstraction and multiple inheritance in Java. Java Interface also represents IS-A relationship.It cannot be instantiated just like abstract class.
There are mainly three reasons to use interface. They are given below.

- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

Pseudo code:

```
interface printable{
void print();
}
class A6 implements printable{
public void print(){System.out.println("Hello");}

public static void main(String args[]){
A6 obj = new A6();
obj.print();
}
}
```

Output:

```
Hello
```

Java Package

A java package is a group of similar types of classes, interfaces and sub-packages.
Package in java can be categorized in two form, built-in package and user-defined package.
There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
Here, we will have the detailed learning of creating and using user-defined packages.
Advantage of Java Package

- Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- Java package provides access protection.
- Java package removes naming collision.

Pseudo code:

```
//save as Simple.java
package mypack;
public class Simple{
public static void main(String args[]){
System.out.println("Welcome to package");
}
}
```

Conclusion: Implementation of concepts of interface and packages is done here.

Questions:-

Q. Multiple inheritance is not supported through class in java but it is possible by interface, why?
Q. How to access package from another package?

ASSIGNMENT NO: 10

AIM: Write a program to implement the concept of exception handling.

OBJECTIVE: Understanding the concepts of exception handling.

BRIEF THEORY:

Exception Handling in Java

he exception handling in java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.

What is exception

Dictionary Meaning: Exception is an abnormal condition.

In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

What is exception handling

Exception Handling is a mechanism to handle runtime errors such as ClassNotFound, IO, SQL, Remote etc.

Advantage of Exception Handling

The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling.

Common scenarios where exceptions may occur

There are given some scenarios where unchecked exceptions can occur. They are as follows:

1) Scenario where ArithmeticException occurs, if we divide any number by zero, there occurs an ArithmeticException.

int a=50/0;//ArithmeticException

2) Scenario where NullPointerException occurs,if we have null value in any variable, performing any operation by the variable occurs an NullPointerException.

String s=null;
System.out.println(s.length());//NullPointerException

3) Scenario where NumberFormatException occurs

The wrong formatting of any value, may occur NumberFormatException. Suppose I have a string variable that have characters, converting this variable into digit will occur NumberFormatException.

String s="abc";
int i=Integer.parseInt(s);//NumberFormatException

4) Scenario where ArrayIndexOutOfBoundsException occurs, if you are inserting any value in the wrong index, it would result ArrayIndexOutOfBoundsException as shown below:

```
int a[]=new int[5];
a[10]=50; //ArrayIndexOutOfBoundsException
```

Conclusion: Implementation of the concepts of exception handling is done here.

Questions:-

Q. Difference between checked and unchecked exceptions.

ASSIGNMENT NO: 11

AIM: Write a program to implement multithreading in Java.

OBJECTIVE: Understanding about multithreading in java.

BRIEF THEORY:

Multithreading in Java

Multithreading in java is a process of executing multiple threads simultaneously.

Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Java Multithreading is mostly used in games, animation etc.

Advantages of Java Multithreading

- It doesn't block the user because threads are independent and you can perform multiple operations at same time.
- You can perform many operations together so it saves time.
- Threads are independent so it doesn't affect other threads if exception occur in a single thread.

Conclusion: Implementation of multithreading in Java is done here.

Questions:-

Q. How to perform multithreading by anonymous class?

ASSIGNMENT NO: 12

AIM: Write a program to implement different graphics object and controls of AWT objects/classes.

OBJECTIVE: Learning about different graphics object and controls of AWT objects/classes.

BRIEF THEORY:

Java AWT

Java AWT (Abstract Window Toolkit) is an API to develop *GUI or* window-based applications in java.Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

The java.awt package provides classes for AWT api such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

Container

The Container is a component in AWT that can contain another components like buttons, textfields, labels etc. The classes that extends Container class are known as container such as Frame, Dialog and Panel.

Window

The window is the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window.

Panel

The Panel is the container that doesn't contain title bar and menu bars. It can have other components like button, textfield etc.

Frame

The Frame is the container that contain title bar and can have menu bars. It can have other components like button, textfield etc.

Pseudo code:

```
import java.awt.*;
class First extends Frame{
First(){
Button b=new Button("click me");
b.setBounds(30,100,80,30);// setting button position
add(b);//adding button into frame
setSize(300,300);//frame size 300 width and 300 height
setLayout(null);//no layout manager
setVisible(true);//now frame will be visible, by default not visible
}
public static void main(String args[]){
First f=new First();
}}
```

Conclusion: Implementation of different graphics object and controls of AWT objects/classes is done here.

Questions:-

Q. What are the various useful methods of Component class?
Q. Use of AWT objects/classes in java.