VETRI VINAYAHA COLLEGE OF ENGINEERING AND TECHNOLOGY
THOTTIAM – 621215
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B.E FOURTH SEMESTER

CS6456 OBJECT ORIENTED PROGRAMMING (Regulation 2013)

UNIT I OVERVIEW

Why Object-Oriented Programming in C++ - Native Types and Statements –Functions and Pointers-Implementing ADTs in the Base Language.

PART - A

1.What is Object Oriented Programming?

Object Oriented Programming is a Programming approach in which there is a collection of objects.

2. What are the Concepts of OOPs

1. Objects
2. Classes
3. Data Abstraction
4. Data Encapsulation
5. Inheritance
6. Polymorphism
7. Message Passing
8. Dynamic Binding

3. Differentiate Procedure Oriented Programming(POP) and Object Oriented Programming(OOP)

POP

1) Emphasis on non-real itrem
2) Programs are divided into functions
3) Data are sharable
4) Structured Programming
5) Top-Down Approach

OOP

1) Emphasis on real item
2) Programs are divided into Objects
3) Data are not sharable
4) Object Oriented Programming
5) Bottom-Up Approach

4. Define Tokens

Smallest individual unit in a program.C++ tokens are Keywords,Identifiers,Constants,Strings,Operators,

5. What are the Data Types in C++

Built-in Data types
User Defined Data types
Derived Data Types

6. Write the Block Structure of C++
Include Files
Class Declaration
Member Function Definitions
Main Function Program

7.What is function? What are the types of Function in C++?
Function is defined as a named group of statements. A function can be called from any point in a program. If a problem can be divided into small modules, then each odule can be represented as a function.
Types of Function
1. Functions with Arguments and No Return Values
2. Functions with No Arguments and No Return Values
3. Functions with Arguments and Return Values

8. What are the Features of Inline Function?
1. Run Faster
2. Function Call & Return is Eliminated
3. Improves Performance

9. What are the Components of Functions?
1. Function Declaration
2. Function Parameters
3. Function Definition
4. Return Statement
5. Function Call

10. What is Default Arguments?
A function with same name, Different arguments is known as Default Arguments

11.What are the parameter passing in C++.
- Pass by value
- Pass by Address
- Pass by reference

12. Define Class?
A class encloses both data and functions that operate on the data, into a single unit.

13. Define Object Based Language.
Object Based Language=Encapsulation + Object Identity
Object Oriented Language= Object Based Features + Inheritance + Polymorphism

14. What are the Access Specifiers in C++.
1. Public
2. Private
3. Protected

15. What is Static Variables?
Defined within the function, static variable initialized only once. Contents of the variables retained throughout the program.

16. Static Member Functions?
Static Function can have accessed by only static members declared in the same class. Static member function called using the name of class instead of its objects.

17.What are the applications of oops?
• Real-time systems.
• Simulation and modeling.
• Object-oriented databases.
• AI and expert systems.

18.Write any four features of OOPS.
• Emphasis is on data rather than on procedure.
• Programs are divided into objects.
• Data is hidden and cannot be accessed by external functions.
• Follows bottom -up approach in program design.

19. Define abstraction and encapsulation.
Wrapping up of data and function within the structure is called as encapsulation.
The insulation of data from direct access by the program is called as data hiding or information binding.
The data is not accessible to the outside world and only those functions, which are wrapped in the class, can access it.

20.Justify the need for static members.
Static variable are normally used to maintain values common to the entire class.
Feature:
• It is initialized to zero when the first object is created. No other initialization is permitted
• only one copy of that member is created for the entire class and is shared by all the objects
• It is only visible within the class, but its life time is the entire class type and scope of each static member variable must be defined outside the class
• It is stored separately rather than objects
Eg: static int count//count is initialized to zero when an object is created.
int classname::count;//definition of static data member

21. Write a C++ program to check whether an integer is a prime or a composite number.
```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int n,i=2,divisor=0;
cout<<"Enter a Number to Find Whether it is Prime or Composite ";
cin>>n;
while(i<n)
{
while(n%i==0)
{ i=i+1;
divisor=divisor+1;
}
i=i+1;
}
if(divisor==0)
```

```
{
cout<<"Number is Prime";
}
else
{
cout<<"Number is Composite"<<endl;

cout<<"Total Divisors except 1 and itself: "<<divisor;
}
getch();
}
```

22.what is an inline function ?

An inline function is a function that is expanded in line when it is invoked. That is compiler replaces the function call with the corresponding function code.

The inline functions are defined as Inline function-header

```
{
function body
}
```

23.What is dynamic binding or late binding?

Binding refers to the linking of a procedure to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at the run-time.

24.What is a default argument ?

Default arguments assign a default value to the parameter, which does not have matching argument in the function call. Default values are specified when the f unction is declared.

Eg : float amount(float principle,int period,float rate=0. 15)

Function call is

Value=amount(5000,7);

Here it takes principle=5000& period=7

And default value for rate=0.15

Value=amount(5000,7,0.34)

Passes an explicit value 0f 0.34 to rate

We must add default value from right to left

25. What are constant arguments ?

keyword is const. The qualifier const tells the compiler that the function should not modify the argument. The compiler will generate an error when this condition is violated. This type of declaration is significant only when we pass arguments by reference or pointers

eg: int strlen(const char *p);

26. How the class is specified ?

Generally class specification has two parts

• class declaration

It describes the type and scope of its member

• class function definition

It describes how the class functions are implemented

The general form is

```
Class class_name
{
private:
```

variable declarations;
function declaration;
public:
variable declaration;
function declaration;
};

27. How to create an object ?
Once the class has been declared, we can create variables of that type by using the classname
Eg:classname x; //memory for x is created

28. How to access a class member ?
object-name. function-name(actual arguments)
eg:x.getdata(100,75.5);

29.What is the difference between a local variable and a data member?
Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight, and cost and uses functions to operate on these attributes.
The attributes are sometimes called as data members because they hold information.

30.What is Pointer Arithmetic?
Pointer Arithmetic is a set of arithmetic operations by which we can use increment(++) or decrement(--) operators using pointers. we can add or subtract an integer to a pointer variable.

31.What is a namespace?
The namespaces are used to group the entities like class, variables, objects, function under a name. The namespaces help to divide the global scope into sub-scopes where each sub scope has its own name. The keyword Using is used for introducing the namespace being used currently.
Eg: **using namespace std;**
All the files in the c++ standard library declare all of its entities while the std namespace.

32.What are storage classes in C++?
The storage classes are used to define the scope of the variable. Various storage class that are allowed in c++ are- auto, register, static and extern.

33. state difference between inline function and macro?
The inline function is analyzed by the compiler whereas the macros are expanded by c++ preprocessor. Macros cannot return any value whereas the inline function can return some value.

34.call by value?
When a function is called, then in that function actual values of the parameters are passed. Compiler executes this type function slowly as values get copied to formal parameters.

35.call by reference?
Reference means use of memory addresses. Calling by reference means passing the reference variable as a parameter to the function. Compiler executes this type function faster as addresses get copied to formal parameters

36.Define ADT?
Abstract data type is a collection of instances and corresponding operations on it. In ADT all the implementation details are hidden.

37. What is abstract class?
The class that does not specify the implementation details, but specifies only what are the operations in the particular data structure.

PART- B

1.What are the basic principles of object oriented programming in c++ explain with neat sketch.
2.What are the various operators available in c++ with neat illustration for each.
3.Summarize various type conversions? Explain each with a program.
4.Write down various data types available in c++ with example.
5.Write about relationship between arrays and pointers.
6.Implementing stack operation using array.
7.what are the statements available in c++?Explain with example.
8.Explain about function overloading with example.

UNIT II
BASIC CHARACTERISTICS OF OOPS

Data Hiding and Member Functions- Object Creation and Destruction- Polymorphism data abstraction: Iterators and Containers.

PART-A

1. **Define function overloading?**
Function overloading means we can use the same function name to create functions that perform a variety of different tasks.
Eg: An overloaded add ( ) function handles different data types as shown below.
// Declarations
i. int add(int a, int b); //add function with 2 arguments of same type
ii. int add(int a, int b, int c); //add function with 3 arguments of same type
iii. double add(int p, double q); //add function with 2 arguments of different type
//Function calls
add (3 , 4); //uses prototype (i.) add
(3, 4, 5); //uses prototype (ii.) add
(3 , 10.0); //uses prototype (iii.)

2.What is data hiding?
The insulation of data from direct access by the program is called as data hiding or information binding.

3.What is data abstraction?
The insulation of data from direct access by the program is called as data hiding or information binding. The data is not accessible to the outside world and only those functions, which are wrapped in the class, can access it.

4.What are data members and member functions?
Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight, and cost and uses functions to operate on these attributes.

The attributes are sometimes called as data members because they hold information. The functions that operate on these data are called as methods or member functions.
Eg: int a,b; // a,b are data members
Void getdata ( ) ; // member function

5.What do you mean by dynamic initialization of variables?
C++ permits initialization of the variables at run-time. This is referred to as dynamic initialization of variables.
In C++ ,a variable can be initialized at run-time using expressions at the place of declaration as,
……..
……
int n =strlen(string);
……..
float area=3.14*rad*rad;
Thus declaration and initialization is done simultaneously at the place where the variable is used for the first time.

6.Define copy constructor?
A copy constructor is used to declare and initialize an object from another object. It takes a reference to an object of the same class as an argument.
Eg: integer i2(i1);
would define the object i2 at the same time initialize it to the values of i1.
Another form of this statement is
Eg: integer i2=i1;
The process of initializing through a copy constructor is known as copy initialization .

7. **What is type conversion?**
Conversion from basic data type to class type can be done in destination class. Using constructors does it. Constructor takes a single argument whose type is to be converted.
Eg: Converting int type to class type
```
class time
{
int hrs,mins;
public:
…….
Time ( int t) //constructor
{
hours= t/60 ; //t in minutes
mins =t % 60;
}
};
```
Constructor will be called automatically while creating objects so that this conversion is done automatically.

8. Explain the functions of default constructor?
The constructor with no arguments is called default constructor
Eg:
```
Class integer
{
int m,n;
Public:
Integer( );
```

…….
};
integer::integer( )//default constructor
{
m=0;n=0;
}
the statement
integer a;
invokes the default constructor

9. What is the need for overloading the assignment operator? .
The **assignment operator** is used to copy the values from one object to another *already existing*
object. The key words here are "already existing". Consider the following example:
1
Cents cMark(5); // calls Cents constructor
2
Cents cNancy; // calls Cents default constructor
3
cNancy = cMark; // calls Cents assignment
operator
In this case, cNancy has already been created by the time the assignment is executed. Consequently, the Cents assignment operator is called. The assignment operator must be overloaded as a member function.

10. Write any four special properties of constructor.
T hey should be declared in the public section
• They are invoked automatically when the objects are created
• They do not have return types, not even void and therefore, and they cannot return values
• They cannot be inherited, though a derived class can call the base class
• They can have default arguments
• Constructors cannot be virtual f unction

11.List any four operators that cannot be overloaded.
• Class member access operator (. , .*)
• Scope resolution operator (::)
• Size operator (sizeof)
• Conditional operator (?:)

12.Define constructor?
A function with the same name as the class itself responsible for construction and returning objects of the class is called constructor.

13.Define destructor?
The function which bears the same name as class itself preceded by ~ is destructor. The destructors automatically called when the object goes out of scope.

14.What is operator overloading?
The process of giving an existing operator a new , additional meaning is called operator overloading.

15. List out the operator which cannot be overloaded ?
1. The dot operator for member access.(.)

2. The dereference member to class operator .(*)
3. Scope resolution operator.
4. Size of operator (sizeof).
5. Conditional ternary operator(:?)
6. Casting operators static_cast<>, dynamic_cast<>, reinterpret_cast<>, const_cast<>
7. # and ## tokens for macro preprocessor.

16. Define new operator?

The operator new is used for allocation of memory, it gets memory from heap. It is similar to malloc() in C.

Ex to get memory for an integer and assign the address of allocated memory to pointer p,

Int* p = new int.

17. Define delete operator?

Delete in C++ does a similar job as free() function in C, i.e it releases the memory occupied by the new operator.

18. Difference between unary and binary operator?

All operators having a single argument are unary operators. When we overload these operators as member function, we do not need to pass any argument explicitly.

Operators with two argument are known as binary operators. They will have a single argument when defined as member function. The first argument to that operator is always the invoking object.

19. Explain insertion and extraction operator?

Insertion operator : The << operator which is used to write to the console is known as insertion operator. It is also known as the output operator.

Extraction operator : The >> operator which is used to read from the keyboard is known as extraction operator. It is also known as the input operator.

20. What is a scope resolution operator?

Scope resolution operator is used to uncover the hidden variables. It also allows access to global version of variables.

Eg:

```
#include<iostream. h>
int m=10; // global variable m
void main ( )
{
int m=20; // local variable m
cout<<"m="<<m<<"\n";
cout<<": : m="<<: : m<<"\n";
}
```

output:

20

10 (: : m access global m)

Scope resolution operator is used to define the function outside the class.

Syntax:

Return type <class name> : : <function name>

Eg:

Void x : : getdata()

21. What is this pointer?

this is an implicit pointer to every member function in a class. This pointer gives the address of the object. Using this pointer we can access the member data in the object.

Syntax:
i) To get address of object
this;
ii) To access member data
this→name of member data;

22.What is a Friend Function?
Friend function is a function that is not a member function of the class but it can access the private and protected members of the class. The friend function is given by a keyword friend.
The general form is
friend data_type function_name( );

PART B

1.Explain the Friend function concept with an example program.
2.What is Dynamic Initialization of objects? Give a program to illustrate your answer.
3.Explain the constructor concept with its types with example programs.
4.Explain the destruction concept with example.
5.Explain the concept of objects as functional arguments.
6.Write a C++ program to manage a bank account using classes and objects.
7.Discuss in detail about polymorphism with example program.

UNIT III
ADVANCED PROGRAMMING
Templates, Generic programming, and STL-Inheritance-Exception-OOP Using C++

PART-A

1.What is a Template?
Templates are generic programming concept. It doesn't depend on any data type. It is generic in nature, it is of two type 1. Function template 2. Class template
1.function template : function templates are generic functions, which work for any data type that is passed to them. The data type is not specified while writing the function. While using the function , we pass the data type and get the required functionality.
2. class template : class template are also generic , whose data members are generic not for specific. While creating objects for that class we can pass data member of our own.

2.Explain the advantages of template?
Templates are used to increase software reusability, efficiency and flexibility. It increases the reusability without inheritance. Function templates in C++ provides generic functions independent of data type. The advantage of class template is that we can define a generic class that works for any data type that is passed to it as a parameter.

3.what is a difference between a function template and template function?
Function template is a generic programming technique in which a framework for the desired functionality is created and using various data types the implantation is used.
Eg:

```
Void add(T &a,T &b)
{
…
…
}
```

The template function is invoking each implementation.

Eg:
add(10,30);
add(11.11,22.22);

4. What is STL?
A collection of generic classes and functions is called the Standard Template Library. STL components are part of C++ standard Library.

5. What are three components of STL
The STL components are containers, algorithms and iterators.

6. Define containers.
Containers are objects that hold data of same type. Containers are divided into three major categories: sequential, associative and derived.

7. What is iterators? What is its characteristic?
An iterator is an object (like a pointer) that points to an element in a container. We can use iterators to move through the contents of containers. Iterators connect algorithms with containers and play a key role in the manipulation of data stored in the containers.

8. What are the best situations for the use of the associative containers
Associate containers are designed to support direct access to elements using keys they are not sequential. Containers are best suited for fast searching, deletion and insertion of data in a structure called tree.

9. What is meant by pure abstract class?
A class containing pure virtual function is called pure abstract class .

10. What are the benefits of inheritance?
1. Code Reuse
2. Ease of code maintenance
3. Increase reliability
4. Improved performance
5. Less maintenance
6. Easy to extension

11. What is meant by pure virtual function?
A virtual function, equated to zero is called a pure virtual function. It is a function declared in a base class that has no definition relative to the base class.

12. What are rules for virtual function.
1. The virtual functions must be members of some class
2. They cannot be static members
3. They are accessed by object pointers
4. Virtual function can be friend of another class.

13. Define Virtual Base Class.
Duplication of inherited members due to multiple paths can be avoided by making the common base class as virtual base class.

14. Define Virtual Function?
A virtual function is a member function that is declared within a base class and redefined by a derived class.

(or)

It is used to invoke exact version of the member function. Virtual functions should be defined in the public section of a class

15. How can you access the virtual functions.
Virtual functions have to be accessed through a pointer to the base class. It is not accessible directly.

16. What is meant by Abstract Class?
It is the one that is not used to create objects. That is, abstract class is designed Only to act as a base class in which at least one pure virtual function is present. It is used to specify an interface that must be implemented by all subclasses.

17. Define Inheritance.
Creating new class from old class. (or) Deriving a new class from old class.

18. What are types of Inheritance?
1. Single Inheritance
2. Multiple Inheritance
3. Multilevel Inheritance
4. Hybrid Inheritance
5. Hierarchical Inheritance

19. What are visibility modes of Inheritance?
1. Private
2. Public
3. Protected

Note: Private members are not inheritable, inaccessible to the objects of derived class.

20. Compare overloading and overriding.

Overloading	Overriding
In function overloading different number of parameters can be passed to the function.	In function overriding the number of parameters that are passed to the function must be same.
The overloading function may have different return type.	The return type of base and derived member function must be the same.
This method allows defining multiple member functions with same name but different signature.	Overriding is a method that allows a class to redefine the behavior of member functions which the derived class inherits from the base class.

21. What is an exception?
Exceptions which occur during the program execution, due to some fault in the input data. Exceptions are run time anomalies or unusual conditions that a program may encounter while executing. It includes divide by zero, access to an array out of its bound, running out of memory.

22. What is the importance of exception handling? What are the basically 3 keywords of exception handling mechanism?
The importance of exception handling is,
1. Dividing the error handling
2. To provide unconditional termination and programmer preferred termination
3. for separating error reporting and error handling
4. To solve the object destroy problem.

The exception-handling mechanism uses three blocks
1)try block – indicating program area where exception can be thrown,
2)throw block – for throwing an exception
3)catch block – actually taking an action for the specific exception.

PART-B

1.Explain virtual function concept with a program to find the distance between two objects.
2.Explain the Inheritance types with example programs.
3.What do u meant by operator overloading? what are its types? Explain the types with eg programs.
4.Expalin single and multiple Inheritance with an example of your own.
5.Describe manipulation of strings with overloading the following operators:<<,>>,+,-

UNIT IV
OVERVIEW OF JAVA

Data types, variables and arrays, operators, control statements, classes, objects, methods – Inheritance

PART-A

1. What is meant by Java?
Object Oriented Multithreaded High Level Programming Language developed by sun Microsystems in 1991.

2. What is meant by Platform?
Platform is the hardware or system software environment in which your program runs. Most platforms are described as a combination of hardware and operating system.

3. Java is platform independent language. Justify.
Platform is the hardware or system software environment in which your program runs. Moreover java language run by any operating system, thats why java is called platform independent languages.

4. What is meant by Java Application? and what are the various applications of Java?
An Application is a program that runs on your computer, under the operating system of that system.
Various applications of java are,

- Applets
- Networking
- Internationalization
- Security
- Object serialization
- Java Database Connectivity (JDBC)

5. What are features of java
1. Simple
2. Object Oriented
3. Distributed
4. Interpreted
5. Robust

6. Secure
7. Architecture-Neutral
8. Portable
9. High Performance
10.Multithreaded
11.Dynamic Language

6. What are features does not supported by java?
1. Goto statement
2. Multiple inheritance
3. Operator overloading
4. Structures and Unions allowed
5. Pointers

7. What are features supported by java
1. Automatic memory management
2. Multithreaded programs

8. Define Java Character Set.
1. Alphabets
2. Digits
3. Special characters

Java uses Unicode character set.

9. What is meant by Java Class Definition?
A java program contains 2 parts. They are
A class definition that encloses the entire program
A main() method that contains the body.
Every java program should contain atleast one class.

10. What is meant by bytecode in java?
Java compiler after compiling the program creates a new file referred as the class file, which contains a special code referred as the bytecode. It is similar to machine language, but unlike machine language, java byte code is exactly the same on every platform.

11. Define JVM(Java Virtual Machine).
It is an abstract computing machine, having an instruction and memory, which is used to implement the java program language. The JVM is responsible for cross platform portability of java.

12.Why java language called as 'robust'?
Java is called robust because due to JVM it can be used on multiple platforms. Secondly it has good mechanism for error checking.

13. What are wrapper class in java?
Wrapper classes are those classes that allow primitive data types to be accessed as objects. The wrapper class is one kind of wrapper around a primitive data type. It represents the primitive data types in its instance of the class.

Primitive data type	Wrapper class
Boolean	java.lang.Boolean
Byte	java.lang.Byte
Char	java.lang.Character

Int	java.lang.Integer

14. Define Garbage Collection in Java?

Garbage Collection also referred as automatic memory management. Periodically frees the memory used by objects that are no longer needed. The garbage collector in java scans dynamic memory areas for objects and marks those that are referenced. After all possible paths to objects are investigated the unreferenced objects are freed.

15. How multiple inheritance is achieved in java?

Java does not support multiple inheritance. It is achieved by the use interface.

16. State the use of super keyword in java and how it is used in java?

The super keyword is used to access a member of an immediate base class, from a derived class.

In inheritance, the keyword super is used for two purposes-

1. to call super class constructor.
2. to call super class method.

17. Mention the various access levels supported in java

1. Public
2. Protected
3. Private

18. Define method overloading.

Java enables 2 or more methods with same name but with different signatures. The signature includes the number of type, and sequence of the arguments passed to a method. The capability to overload a method is referred to as overloading methods.

PART-B

1.Explain Method overriding in Java with an e.g.

2.Construct pictorial representation of Java Virtual Machine.

3.Write a program in java using constructor concept.

4.What are the different statements and its use in java?

5. Illustrate Inheritance in Java with suitable program.

6.Give a explanatory answer to define the difference between Java and C++,Characteristics of Java and the concepts in java.

UNIT V

EXCEPTION HANDLING

Packages and Interfaces, Exception handling, Multithreaded programming, Strings, Input/Output

PART-A

1.What are packages?

A Java package is a mechanism for organizing Java classes into namespaces similar to the modules of Modula. Java packages can be stored in compressed files called JAR files, allowing classes to download faster as a group rather than one at a time.

2. List out the Java packages.

The Java packages are,

i. java.lang
ii. java.io
iii. java.awt
iv. java.net
v. java.applet
vi. java.rmi

3.What are Encapsulation, Inheritance and Polymorphism?

i. Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse.
ii. Inheritance is the process by which one object acquires the properties of another object.
iii. Polymorphism is the feature that allows one interface to be used for general class actions.

4.What is the use of super keyword?

A Java(TM) programming language keyword used to access members of a class inherited by the class in which it appears.

5.What do mean by overriding methods?

The ability of a subclass to override a method in its superclass allows a class to inherit from a superclass whose behavior is "close enough" and then supplement or modify the behavior of that superclass.

6.List out the various types of inheritance.

The various types of inheritance are,

i. Simple inheritance - One base class and one derived class
ii. Multilevel inheritance - a sub class derived from another sub class
iii. Hierarchical inheritance - two or more sub class have the same base class

7.Define Interface.

An Interface is defined as a named collection of method definitions (without implementations). An interface can also declare constants. All the methods declared in the interface are abstract methods by default. And all the data members are static final members.

8. How multiple inheritance is implemented in java?

A class can implement more than one interface (the Java platform supports multiple inheritance for interfaces), so the `implements` keyword is followed by a comma-separated list of the interfaces implemented by the class.

9. What is an Exception?

An Exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions during the execution of a program.

10. How an Exception is handled in java?

A program can catch exceptions by using a combination of the try, catch, and finally statements. The try statement identifies a block of code in which an exception can occur. The catch statement identifies a block of code, known as an exception handler that can handle a particular type of exception.

11. What is the use of finally block?

The finally statement identifies a block of code that cleans up regardless of whether an exception occurred within the try block. A try statement must be accompanied by at least one catch statement or a finally statement and may have multiple catch statements.

12. How Threads are created in Java?

Threads are created in two ways,

i. extending the Thread class
ii. implementing Runnable interface.

13.What is Thread pool?

A Thread pool is a managed collection of threads that are available to perform tasks. Thread pools usually provide,

i. Improved performance when executing large numbers of tasks due to reduced per-task invocation overhead.
ii. A means of bounding the resources, including threads, consumed when executing a collection of tasks.

14. Define Dead Lock.

A Dead Lock is defined as a special type of error that relates specifically to multitasking is dead lock, which occurs when two threads have a circular dependency on a pair of synchronized objects.

15. What do you mean by Thread Scheduling?

Execution of multiple threads on a single CPU in some order is called Thread scheduling. The Java runtime environment supports a very simple, deterministic scheduling algorithm called *fixed-priority scheduling*. This algorithm schedules threads on the basis of their priority relative to other Runnable threads.

16. What is the use of Data input and output streams?

Data input and output Streams are used to read or write primitive data types in a machine-independent format.

17. What is the use of Buffered Streams?

Buffered streams, buffer data while reading or writing, thereby reducing the number of accesses required on the original data source. Buffered streams are typically more efficient than similar non buffered streams and are often used with other streams.

18. What is filtered Stream?

A filtered stream filters data as its being read from or written to the stream. The java.io package provides a set of abstract classes that define and partially implement filter streams.

19.Name two super classes used in byte stream.

The InputStream and OutputStream classes.

PART B

1.Explain the Life cycle of Thread with an e.g.
2.Describe package concept to perform arithmetic operations. Explain how to use it.
3.Explain the different states in Life cycle of applet?
4.Define Interfaces?Explain the extension of interfaces, implementation and accessing it.
6.Explain try, catch and finally statements with e.g.