1. **What is method overriding in java?**

If we have methods with same signature (same name, same signature, same return type) in super class and subclass then we say subclass method is overridden by superclass.

When to use overriding in java

If we want same method with different behavior in superclass and subclass then we go for overriding. When we call overridden method with subclass reference subclass method is called hiding the superclass method.

1. **What is super keyword in java ?**

Variables and methods of super class can be overridden in subclass . In case of overriding , a subclass object call its own variables and methods. Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still java provides a way to access super class members even if its members are overridden. Super is used to access superclass variables, methods, constructors.

Super can be used in two forms :

1. First form is for calling super class constructor.
2. Second one is to call super class variables, methods. Super if present must be the first statement.

1. **Difference between method overloading and method overriding in java ?**

|  |  |
| --- | --- |
| **Method Overloading** | **Method Overriding** |
| 1) Method Overloading occurs with in the same class | Method Overriding occurs between two classes superclass and subclass |
| 2) Since it involves with only one class inheritance is not involved. | Since method overriding occurs between superclass and subclass inheritance is involved. |
| 3)In overloading return type need not be the same | 3) In overriding return type must be same. |
| 4) Parameters must be different when we do overloading | 4) Parameters must be same. |
| 5) Static polymorphism can be acheived using method overloading | 5) Dynamic polymorphism can be acheived using method overriding. |
| 6) In overloading one method can’t hide the another | 6) In overriding subclass method hides that of the superclass method. |

1. **Difference between abstract class and interface ?**

|  |  |
| --- | --- |
| Interface | Abstract Class |
| 1) Interface contains only abstract methods | 1) Abstract class can contain abstract methods, concrete methods or both |
| 2) Access Specifiers for methods in interface must be public | 2) Except private we can have any access specifier for methods in abstract class. |
| 3) Variables defined must be public , static , final | 3) Except private variables can have any access specifiers |
| 4) Multiple Inheritance in java is implemented using interface | 4)We cannot achieve multiple inheritance using abstract class. |
| 5) To implement an interface we use implements keyword | 5)To implement an interface we use implements keyword |

1. **What is method overloading in java ?**

A class having two or more methods with same name but with different arguments then we say that those methods are overloaded. Static polymorphism is achieved in java using method overloading.

Method overloading is used when we want the methods to perform similar tasks but with different inputs or values. When an overloaded method is invoked java first checks the method name, and the number of arguments ,type of arguments; based on this compiler executes this method.

Compiler decides which method to call at compile time. By using overloading static polymorphism or static binding can be achieved in java.

Note : Return type is not part of method signature. we may have methods with different return types but return type alone is not sufficient to call a method in java.

1. **What is bytecode in java ?**

When a javac compiler compiles a class it generates .class file. This .class file contains set of instructions called byte code. Byte code is a machine independent language and contains set of instructions which are to be executed only by JVM. JVM can understand this byte codes.

1. **Difference between this() and super() in java ?**

this() is used to access one constructor from another with in the same class while super() is used to access superclass constructor. Either this() or super() exists it must be the first statement in the constructor.

1. **What is a class ?**

Classes are fundamental or basic unit in Object Oriented Programming .A class is kind of blueprint or template for objects. Class defines variables, methods. A class tells what type of objects we are creating. For example take Department class tells us we can create department type objects. We can create any number of department objects.

All programming constructs in java reside in class. When JVM starts running it first looks for the class when we compile. Every Java application must have at least one class and one main method.

Class starts with class keyword. A class definition must be saved in class file that has same as class name. File name must end with .java extension.

**public** **class** FirstClass

{**public** **static** **void** main(String[] args)

{System.*out*.println(“My First class”);

}

}

If we see the above class when we compile JVM loads the FirstClass and generates a .class file(FirstClass.class). When we run the program we are running the class and then executes the main method.

1. **What is an object ?**

An Object is instance of class. A class defines type of object. Each object belongs to some class.Every object contains state and behavior. State is determined by value of attributes and behavior is called method. Objects are alos called as an instance. To instantiate the class we declare with the class type.

**public** **class**FirstClass {**public** **static** **void**main(String[] args) {

FirstClass f=new FirstClass();

System.*out*.println(“My First class”);

}

}

To instantiate the FirstClass we use this statement FirstClass f=new FirstClass(); f is used to refer FirstClass object.

1. **What is method in java ?**

It contains the executable body that can be applied to the specific object of the class.

Method includes method name, parameters or arguments and return type and a body of executable code.

Syntax : type methodName(Argument List){

}

ex : public float add(int a, int b, int c) methods can have multiple arguments. Separate with commas when we have multiple arguments.

1. **What is encapsulation ?**

*The process of wrapping or putting up of data in to a single unit class and keeps data safe from misuse is called encapsulation .*

Through encapsulation we can hide and protect the data stored in java objects.Java supports encapsulation through access control. There are four access control modifiers in java public , private ,protected and default level.

For example take a car class , In car we have many parts which is not required for driver to know what all it consists inside. He is required to know only about how to start and stop the car. So we can expose what all are required and hide the rest by using encapsulation.

1. **Why main() method is public, static and void in java ?**

public : “public” is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.

static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.

void : void return type is used when a method doesn’t return any value . main() method doesn’t return any value, so main() is declared as void.

Signature : public static void main(String[] args) {

1. **Explain about main() method in java ?**

Main() method is starting point of execution for all java applications.

public static void main(String[] args) {}

String args[] are array of string objects we need to pass from command line arguments. Every Java application must have at least one main method.

1. **What is constructor in java ?**

*A constructor is a special method used to initialize objects in java.*

we use constructors to initialize all variables in the class when an object is created. As and when an object is created it is initialized automatically with the help of constructor in java.

We have two types of constructors

Default Constructor

Parameterized Constructor

Signature : public classname()

{

}

Signature : public classname(parameters list)

{

}

1. **What is difference between length and length() method in java ?**

length() : In String class we have length() method which is used to return the number of characters in string.

Ex : String str = “Hello World”;

System.out.println(str.length());

Str.length() will return 11 characters including space.

length : we have length instance variable in arrays which will return the number of values or objects in array. For example :

String days[]={” Sun”,”Mon”,”wed”,”thu”,”fri”,”sat”}; Will return 6 since the number of values in days array is 6.

1. **What is ASCII Code?**

ASCII stands for American Standard code for Information Interchange. ASCII character range is 0 to 255.

We can’t add more characters to the ASCII Character set. ASCII character set supports only English. That is the reason, if we see C language we can write c language only in English we can’t write in other languages because it uses ASCII code.

1. **What is Unicode ?**

Unicode is a character set developed by Unicode Consortium. To support all languages in the world Java supports Unicode values. Unicode characters were represented by 16 bits and its character range is 065,535.

Java uses ASCII code for all input elements except for Strings,identifiers, and comments. If we want to use telugu we can use telugu characters for identifiers. We can enter comments in telugu.

1. **What is ‘IS-A ‘ relationship in java?**

‘is a’ relationship is also known as inheritance. We can implement ‘is a’ relationship or inheritance in java using extends keyword. The advantage or inheritance or is a relationship is reusability of code instead of duplicating the code.

Ex : Motor cycle is a vehicle

Car is a vehicle Both car and motorcycle extends vehicle.

1. **What is ‘HAS A’’ relationship in java?**

‘Has a ‘ relationship is also known as “composition or Aggregation”. As in inheritance we have ‘extends’ keyword we don’t have any keyword to implement ‘Has a’ relationship in java. The main advantage of ‘Has-A‘ relationship in java code reusability.

1. **Difference between ‘IS-A’ and ‘HAS-A’ relationship in java?**

|  |  |
| --- | --- |
| IS-A relationship | HAS- A RELATIONSHIP |
| Is a relationship also known as inheritance | Has a relationship also known as composition or aggregation. |
| For IS-A relationship we uses extends keyword | For Has a relationship we use new keyword |
| Ex : Car is a vehicle. | Ex : Car has an engine. We cannot say Car is an engine |
| The main advantage of inheritance is reusability of code | The main advantage of has a relationship is reusability of code. |

1. **Explain about instanceof operator in java?**

Instanceof operator is used to test the object is of which type.

Syntax : <reference expression> instanceof <destination type>

Instanceof returns true if reference expression is subtype of destination type. Instanceof returns false if reference expression is null.

|  |
| --- |
| Example : **public** **class**InstanceOfExample {**public** **static** **void**main(String[] args) {Integer a = **new**Integer(5);**if** (a **instanceof** java.lang.Integer) {  System.*out*.println(**true**);  } **else** {  System.*out*.println(**false**); } |

}

}

Since a is integer object it returns true.

There will be a compile time check whether reference expression is subtype of destination type. If it is not a subtype then compile time error will be shown as Incompatible types

1. **What does null mean in java?**

When a reference variable doesn’t point to any value it is assigned null.

Example : Employee employee;

In the above example employee object is not instantiate so it is pointed no where

1. **Can we have multiple classes in single file ?**

Yes we can have multiple classes in single file but it people rarely do that and not recommended. We can have multiple classes in File but only one class can be made public. If we try to make two classes in File public we get following compilation error.

“The public type must be defined in its own file”.

1. **What are packages in java?**

Package is a mechanism to group related classes ,interfaces and enums in to a single module. Package can be declared using the following statement :

Syntax : package <package-name>

Coding Convention : package name should be declared in small letters. package statement defines the namespace.

The main use of package is

1. To resolve naming conflicts
2. For visibility control : We can define classes and interfaces that are not accessible outside the class.

1. **Can we have more than one package statement in source file ?**

We can’t have more than one package statement in source file. In any java program there can be atmost only 1 package statement. We will get compilation error if we have more than one package statement in source file.

1. **Can we define package statement after import statement in java?**

We can’t define package statement after import statement in java. package statement must be the first statement in source file. We can have comments before the package statement.

1. **What are identifiers in java?**

Identifiers are names in java program. Identifiers can be class name, method name or variable name.

Rules for defining identifiers in java:

1. Identifiers must start with letter, underscore or dollar($) sign.
2. Identifiers can’t start with numbers .
3. There is no limit on number of characters in identifier but not recommended to have more than 15 characters
4. Java identifiers are case sensitive.
5. First letter can be alphabet ,or underscore and dollar sign. From second letter we can have numbers

.

1. We shouldn’t use reserve words for identifiers in java.

1. **What are access modifiers in java?**

The important feature of encapsulation is access control. By preventing access control we can misuse of class, methods and members.

A class, method or variable can be accessed is determined by the access modifier. There are three types of access modifiers in java. public,private,protected. If no access modifier is specified then it has a default access.

1. **Explain about abstract classes in java?**

Sometimes we may come across a situation where we cannot provide implementation to all the methods in a class. We want to leave the implementation to a class that extends it. In such case we declare a class as abstract. To make a class abstract we use key word abstract. Any class that contains one or more abstract methods is declared as abstract. If we don’t declare class as abstract which contains abstract methods we get compile time error. We get the following error.

*“The type <class-name> must be an abstract class to define abstract methods.”*

Signature ; abstract class <class-name>

{

}

For example if we take a vehicle class we cannot provide implementation to it because there may be two wheelers , four wheelers etc. At that moment we make vehicle class abstract. All the common features of vehicles are declared as abstract methods in vehicle class. Any class which extends vehicle will provide its method implementation. It’s the responsibility of subclass to provide implementation.

The important features of abstract classes are :

1. Abstract classes cannot be instantiated.
2. An abstract classes contains abstract methods, concrete methods or both.
3. Any class which extends abstract class must override all methods of abstract class.
4. An abstract class can contain either 0 or more abstract methods.

Though we cannot instantiate abstract classes we can create object references . Through superclass references we can point to subclass.

1. **Can we create constructor in abstract class ?**

We can create constructor in abstract class , it doesn’t give any compilation error. But when we cannot instantiate class there is no use in creating a constructor for abstract class.

1. **What are abstract methods in java?**

An abstract method is the method which doesn’t have any body. Abstract method is declared with keyword abstract and semicolon in place of method body.

Signature : public abstract void <method name>();

Ex : public abstract void getDetails();

It is the responsibility of subclass to provide implementation to abstract method defined in abstract class. **Java Exception Handling Interview questions**

1. **What is an exception in java?**

In java exception is an object. Exceptions are created when an abnormal situations are arised in our program. Exceptions can be created by JVM or by our application code. All Exception classes are defined in java.lang. In other words we can say Exception as run time error.

1. **What is Exception handling in java?**

Exception handling is a mechanism what to do when some abnormal situation arises in program. When an exception is raised in program it leads to termination of program when it is not handled properly. The significance of exception handling comes here in order not to terminate a program abruptly and to continue with the rest of program normally. This can be done with help of Exception handling.

1. **What is an error in Java?**

Error is the subclass of Throwable class in java. When errors are caused by our program we call that as Exception, but some times exceptions are caused due to some environment issues such as running out of memory. In such cases we can’t handle the exceptions. Exceptions which cannot be recovered are called as errors in java.

Ex : Out of memory issues.

1. **Explain try and catch keywords in java?**

In try block we define all exception causing code. In java try and catch forms a unit. A catch block catches the exception thrown by preceding try block. Catch block cannot catch an exception thrown by another try block. If there is no exception causing code in our program or exception is not raised in our code jvm ignores the try catch block.

Syntax :

try {

}

Catch(Exception e)

{

}

1. **What is process ?**

***A process is a program in execution.***

Every process have their own memory space. Process are heavy weight and requires their own address space. One or more threads make a process.

1. **What is thread in java?**

***Thread is separate path of execution in program.***

Threads are

1. Light weight
2. They share the same address space.
3. creating thread is simple when compared to process because creating thread requires less resources when compared to process

Threads exists in process. A process have at least one thread.

1. **What is multitasking ?**

Multitasking means ***performing more than one activity at a time***on the computer. Example Using spreadsheet and using calculator at same time.

1. **Explain thread in java?**
2. Thread is independent path of execution with in a program.
3. A thread consists of three parts Virtual Cpu, Code and data.

At run time threads share code and data i.e they use same address space. 4) Every thread in java is an object of java.lang.Thread class.

1. **What are synchronized methods ?**

If we want a method of object to be accessed by single thread at a time we declare that method with synchronized keyword. Signature :

**public** **synchronized** **void** methodName(){}

To execute synchronized method first lock has to be acquired on that object. Once synchronized method is called lock will be automatically acquired on that method when no other thread has lock on that method. once lock has been acquired then synchronized method gets executed. Once synchronized method execution completes automatically lock will be released. The prerequisite to execute a synchronized method is to acquire lock before method execution. If there is a lock already acquired by any other thread it waits till the other thread completes.

1. **What are thread groups?**

Thread Groups are group of threads and other thread groups. It is a way of grouping threads so that actions can be performed on set of threads for easy maintenance and security purposes.

For example we can start and stop all thread groups. We rarely use thread group class. By default all the threads that are created belong to default thread group of the main thread. Every thread belongs to a thread group. Threads that belong to a particular thread group cannot modify threads belonging to another thread group.

1. **What are thread local variables ?**

Thread local variables are variables associated to a particular thread rather than object. We declare ThreadLocal object as private static variable in a class. Every time a new thread accesses object by using getter or setter we are accessing copy of object. Whenever a thread calls get or set method of ThreadLocal instance a new copy is associated with particular object.

1. **Difference between object and reference?**

Reference and object are both different. Objects are instances of class that resides in heap memory. Objects doesn’t have any name so to access objects we use references. There is no alternative way to access objects except through references.

Object cannot be assigned to other object and object cannot be passed as an argument to a method. Reference is a variable which is used to access contents of an object. A reference can be assigned to other reference ,passed to a method.

1. **Explain wrapper classes in java?**

Converting primitives to objects can be done with the help of wrapper classes. Prior to java 1.5 we use Wrapper classes to convert primitives to objects. From java 1.5 we have a new feature autoboxing which is used to convert automatically primitives to objects but in wrapper classes programmer has to take care of converting primitives to objects.

Wrapper classes are immutable in java. Once a value is assigned to it we cannot change the value.

1. **What is classpath ?**

The path where our .class files are saved is referred as classpath.JVM searches for .class files by using the class path specified. Class path is specified by using CLASSPATH environment variable. CLASSPATH environment variable can contain more than one value. CLASSPATH variable containing more than one value is separated by semicolon.

Example to set class path from command prompt :

set CLASSPATH= C:Program FilesJavajdk1.6.0\_25bin;.;

only parent directories need to be added to classpath.Java compiler will look for appropriate packages and classes.

1. **What is jar ?**

Jar stands for java archive file. Jars are created by using Jar.exe tool. Jar files contains .class files, other resources used in our application and manifest file. Manifest file contains class name with main method.jar contains compressed .class files. Jvm finds these .class files without uncompressing this jar.

1. **Define interface in java?**

Interface is collection of abstract methods and constants. An interface is also defined as pure or 100 percent abstract class. Interfaces are implicitly abstract whether we define abstract access modifier or not. A class implementing interface overrides all the abstract methods defined in interface. Implements keyword is used to implement interface.

1. **Explain about object oriented programming and its features?**

Java replaced traditional programming language developed in 1970’s. In Object oriented programming everything is made up of object. In this language bottom up approach is followed. Each object communicates with other as opposed to traditional view.

Features :

1. In this bottom approach is followed. First concentrates on minute details like creating objects then concentrates on implementation or solving the problem.
2. Concentrate more on data and give less importance for implementation.
3. Objects communicate with each other

The main advantage of object oriented programming language is works well for larger problems.

1. **Explain what is encapsulation?**

Encapsulation is the process of wrapping of code and behavior in a single unit called class and preventing from misuse is called encapsulation. Encapsulation exposes only part of object which are safe to exposed and remaining part of object is kept secured.

Encapsulation is supported through access control in java. There are four types of access control specifiers(public, private, protected, default) in java which supports encapsulation.

For example tv manufacturers exposes only buttons not all the thousands of electronic components which it is made up of.

1. **What is inheritance ?**

Inheritance is one of the important feature of object oriented language. Inheriting is the process of acquiring features of others. For example a child acquires the features of their parents.

In java inheritance is the process of inheriting member of existing classes by extending their functionality. The original class is called base class, parent class or super class. The new class derived from parent is called child class, sub class, and derived class.

We use extends keyword in java to extend a class in java. All java classes extend java.lang.Object since object class is the super class for all classes in java.

When we create a new class by using inheritance ‘is-a’ relationship is formed.

1. **What is collection ?**

A collection is a container which holds group of objects. Collection provides a way to manage objects easily. Collections manages group of objects as single unit.

Examples include list of strings, integers etc.

Here are few basic operations we do on collections :

1. Adding objects to collection.
2. Removing or deleting objects from collection.
3. Retrieving object from collection.
4. Iterating collection.
5. **Explain List interface ?**

List interface extends collection interface used to store sequence of elements in collection.

We can even store duplicate elements in list.

We can insert or access elements in list by using index as we do in arrays.

List is an ordered collection.

The main difference between List and non list interface are methods based on position.

Some of the operations we can perform on List :

1. Adding an element at specified index.
2. Removing an element at specified index.
3. To get the index of element

List contains some specific methods apart from Collection interface methods.

1. **Explain about ArrayList ?**

ArrayList is an ordered collection which extends AbstractList and implements List interface. We use ArrayList mainly when we need faster access and fast iteration of elements in list.

We can insert nulls in to arraylist.

Arraylist is nothing but a growable array.

**public class ArrayList<E> extends AbstractList<E> implements List<E>, RandomAccess, Cloneable, java.io.Serializable{}**

From java 1.4 ArrayList implements RandomAccess interface which is a marker interface which supports fast and random access.

Advantages :

1. Faster and easier access.
2. Used for Random access of elements.

Drawbacks :

1) We cannot insert or delete elements from middle of list.

1. **What is vector?**

Vector is similar to arraylist used for random access.

Vector is a dynamic array like arraylist.

vector size increases or decreases when elements are added and removed .

Vector is synchronized .

vector and Hashtable are the only collections since 1.0. Rest of the collections are added from 2.0.

public class Vector<E>extends AbstractList<E>implements List<E>,

RandomAccess, Cloneable, java.io.Serializable

1. **Explain about Sets ?**

A set is a collection which does not allow duplicates. Set internally implements equals() method which doesn’t allow duplicates. Adding an duplicate element to a set would be ignored .Set interface is implemented in java.util.set package. Set interface does not have any additional methods . It has only collection methods. A set can contain at most one null value.

ArrayList is an ordered collection. In arraylists order remains same in which they are inserted. But coming to set it is an unordered collection.

public interface Set<E> extends Collection<E> {

}

Important operations that can be performed on set :

1. Adding an element to set.
2. Removing an element from set.
3. Check if an element exist in set.

Iterating through set.

1. **Explain about Map interface in java?**

A map is an association of key-value pairs. Both keys and values in map are objects.

Features of map :

1) Maps cannot have duplicate keys but can have duplicate value objects.

1. **What is serialization in java?**

Serialization is the process of converting an object in to bytes, so that it can be transmitted over the network,or stored in a flat file and can be recreated later. Serialized object is an object represented as sequence of bytes that includes objects data, object type, and the types of data stored in the object.

1. **What is the main purpose of serialization in java?**

The main uses of serialization are :

1. Persistence:

We can write data to a file or database and can be used later by deserializing it.

1. Communication :

To pass an object over network by making remote procedure call.

1. Copying :

We can create duplicates of original object by using byte array. 4) To distribute objects across different JVMs.

1. **Explain about serializable interface in java?**

To implement serialization in java there is an interface defined in java.io package called serializable interface. Java.io.Serializable interface is an marker interface which does not contain any methods. A class implements Serializable lets the JVM know that the instances of the class can be serialized.

Syntax:

public interface Serializable {

}

1. What are identifiers?

*Identifiers are the names that identify the elements such as classes, methods, and variables in a program.*

1. *What are variables?*

*Variables are used to represent values that may be changed in the program.*

1. *What is an assignment statement?*

*An assignment statement designates a value for a variable. An assignment statement can be used as an expression in Java.*

1. *What is a named constant?*

*A named constant is an identifier that represents a permanent value.*

1. *What is a literal?*

*A* literal *is a constant value that appears directly in a program.*

1. *What is the software development life cycle?*

*The software development life cycle is a multistage process that includes requirements specification, analysis, design, implementation, testing, deployment, and maintenance.*

1. *What is an* **if** *statement?*

*An* **if** *statement is a construct that enables a program to specify alternative paths of execution.*

1. *What is an* **if-else** *statement?*

*An* **if-else** *statement decides the execution path based on whether the condition is true or false.*

1. *What is a* **switch** *statement?*

*A* **switch** *statement executes statements based on the value of a variable or an expression.*

1. *What is Debugging?*

*Debugging is the process of finding and fixing errors in a program.*

1. *What is a loop?*

*A loop can be used to tell a program to execute statements repeatedly.*

1. *What is a* **while** *loop?*

*A* **while** *loop executes statements repeatedly while the condition is true.*

1. *What is a* **do-while** *loop?*

*A* **do-while** *loop is the same as a* **while** *loop except that it executes the loop body first and then checks the loop continuation condition.*

1. *What is a method?*

*A method definition consists of its method name, parameters, return value type, and body.*

1. *What are Overloading methods?*

*Overloading methods enables you to define the methods with the same name as long as their signatures are different.*

1. *Explain array?*

*A single array variable can reference a large collection of data. Once an array is created, its size is fixed. An array reference variable is used to access the elements in an array using an* index*.*

1. *What is a class?*

*A class defines the properties and behaviors for objects.* A *class* is a template for *objects*. It defines the *properties* of objects and provides *constructors* for creating objects and methods for manipulating them. A class is also a data type.

1. *What is an object?*

An *object* represents an entity in the real world that can be distinctly identified. The *state* of an object (also known as its *properties* or *attributes*) is represented by *data fields* with their current values. The *behavior* of an object (also known as its *actions*) is defined by methods.

*Classes are definitions for objects and objects are created from classes.*

*The contents of immutable objects cannot be changed.*

1. *What does* **this** *keyword means?*

*The keyword* **this** *refers to the object itself. It can also be used inside a constructor to invoke another constructor of the same class.*

1. What is class *abstraction?*

*Class abstraction is the separation of class implementation from the use of a class.*

1. What is *class encapsulation?*

*The details of implementation are encapsulated and hidden from the user. This is known as class encapsulation.*

1. *What does* **super** *keyword means?*

*The keyword* **super** *refers to the superclass and can be used to invoke the superclass’s methods and constructors.*

1. *How to override a method?*

*To override a method, the method must be defined in the subclass using the same signature and the same return type as in its superclass.*

1. What is *Polymorphism?*

*Polymorphism means that a variable of a supertype can refer to a subtype object.*

1. What is an **ArrayList?**

*An* **ArrayList** *object can be used to store a list of objects.*

1. What is an *exception?*

An *exception* is an object that represents an error or a condition that prevents execution from proceeding normally.

*Use the* **Scanner** *class for reading text data from a file and the* **PrintWriter** *class for writing text data to a file.*

1. What is an *abstract class?*

*An abstract class cannot be used to create objects. An abstract class can contain abstract methods, which are implemented in concrete subclasses.*

1. What is an *interface?*

*An interface is a class-like construct that contains only constants and abstract methods. A class can implement multiple interfaces, but it can only extend one superclass.*

1. What is an *event?*

*An event is an object created from an event source. Firing an event means to create an event and delegate the handler to handle the event.*

1. What is a *handler?*

*A handler is an object that must be registered with an event source object, and it must be an instance of an appropriate event-handling interface.*

1. What is a *recursive method?*

*A recursive method is one that invokes itself. Sometimes you can find a solution to the original problem by defining a recursive function to a problem similar to the original problem. This new method is called a recursive helper method. The original problem can be solved by invoking the recursive helper method.*

1. What is *Recursion?*

*Recursion is an alternative form of program control. It is essentially repetition without a loop.*

1. What is *Generics?*

*Generics enable you to detect errors at compile time rather than at runtime. A generic type can be defined for a class or interface. A concrete type must be specified when using the class to create an object or using the class or interface to declare a reference variable.*

1. What is *The* **Collection** *interface defines?*

*The* **Collection** *interface defines the common operations for lists, vectors, stacks, queues, priority queues, and sets.*

1. *What are maps?*

Maps are efficient data structures for quickly searching an element using a key. *A map is like a dictionary that provides a quick lookup to retrieve a value using a key.*

**Set**s store a group of nonduplicate elements.

**List**s store an ordered collection of elements.

**Stack**s store objects that are processed in a last-in, first-out fashion.

**Queue**s store objects that are processed in a first-in, first-out fashion.

**PriorityQueue**s store objects that are processed in the order of their priorities.

**Iterator** is a classic design pattern for walking through a data structure without having to expose the details of how data is stored in the data structure.

1. What is the list?

*The* **List** *interface extends the* **Collection** *interface and defines a collection for storing elements in a sequential order. To create a list, use one of its two concrete classes:* **ArrayList** *or* **LinkedList***.*

*In a priority queue, the element with the highest priority is removed first.*

1. *What is a set?*

*A set is an efficient data structure for storing and processing nonduplicate elements.*

1. *What is sets?*

*Sets are more efficient than lists for storing nonduplicate elements. Lists are useful for accessing elements through the index.*

1. *What is Algorithm?*

*Algorithm design is to develop a mathematical process for solving a problem. Algorithm analysis is to predict the performance of an algorithm.*

1. *What is insertion-sort algorithm?*

*The insertion-sort algorithm sorts a list of values by repeatedly inserting a new element into a sorted sublist until the whole list is sorted.*

1. *What is a bubble sort?*

*A bubble sort sorts the array in multiple phases. Each pass successively swaps the neighboring elements if the elements are not in order.*

1. *What is the merge sort algorithm?*

*The merge sort algorithm can be described recursively as follows: The algorithm divides the array into two halves and applies a merge sort on each half recursively. After the two halves are sorted, merge them.*

1. *What is a quick sort?*

*A quick sort works as follows: The algorithm selects an element, called the pivot, in the array. It divides the array into two parts, so that all the elements in the first part are less than or equal to the pivot and all the elements in the second part are greater than the pivot. The quick sort algorithm is then recursively applied to the first part and then the second part.*

1. *What is a heap sort?*

*A heap sort uses a binary heap. It first adds all the elements to a heap and then removes the largest elements successively to obtain a sorted list.*

*What are Stacks?*

*Stacks can be implemented using array lists and queues can be implemented using linked lists.*

1. *What is Hashing?*

*Hashing uses a hashing function to map a key to an index.*

1. *What is Multithreading?*

*Multithreading enables multiple tasks in a program to be executed concurrently.*

1. *What is a thread?*

*A program may consist of many tasks that can run concurrently. A thread is the flow of execution, from beginning to end, of a task. The* **Thread** *class contains the constructors for creating threads for tasks and the methods for controlling threads.*

1. *What is a task?*

*A task class must implement the* **Runnable** *interface. A task must be run from a thread. Thread synchronization is to coordinate the execution of the dependent threads. A thread state indicates the status of thread.*