

- Q Define ADTs, what are the importance of ADT and what are the role of encapsulation and information hiding.

Sol:-

ADT are logical models that define data structure by specifying the operations that can be performed on it and their behavior, without focusing on how they are implemented.  
Importance:-

\* ADTs combine data and operations, hiding the internal details.

- \* They can be reused in different projects.
- \* ADTs help divide programs into smaller, manageable parts.
- \* Changes in implementation don't affect how they are used.
- \* Focus is on functionality, not implementation.
- \* ex:- ~~stack~~ in stacks, queues, and lists

### Role of encapsulation and

- ② Encapsulation groups data and operations into a single unit, ensuring controlled access through defined methods.

Information hiding:-

controlled internal implementation, exposing only necessary functionality, improving security, reducing complexity, and supporting maintainability and reusability.

- ③ Explain "Rep-invariant and abstraction function" in ADT.

Sol:- Representation invariant:-

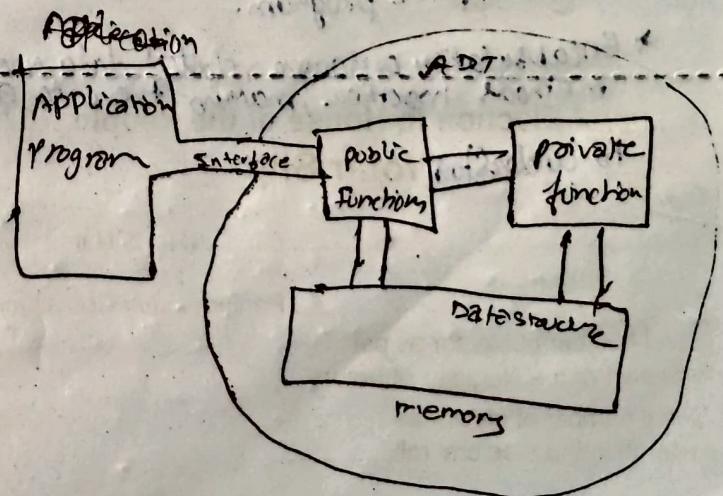
- \* Ensures the internal state of an ADT follows specific rules.
- \* Prevents the ADT from entering invalid states.
- \* Defines constraints on internal data structure.
- \* ex:- in a stack, the array used must be non-null.

abstraction function

- \* connects the internal representation to the abstract concept.
  - \* Translates internal data into a usable form.
  - \* Helps make the abstract
  - \* Ex:- In stack, it represents the array elements as a logical stack.
- Q Define the concept of Polymorphism in Java and discuss its significance. How can you implement Polymorphism through method overloading and interfaces in Java? Provide a Java code example illustrating polymorphism.

Sol:- Java Polymorphism means "many forms." It allows the same method to behave differently depending on the object that calls it.

- \* In Java, Polymorphism lets one method have different behaviours for different objects.
- \* Significance of polymorphism
- \* Code Reusability:- You can use the same method name for different tasks.
- \* It lets you use objects of different classes in the same way.
- \* It makes your code easier to change and update.
- \* It also used to reduce complexity.



# General Election to House of the People, 2024

## Voter Slip

- ④ Define the concept of encapsulation in object-oriented programming. Explain how encapsulation helps in organizing and protecting within the objects. Provide an example in column 1 of the Register of Voters.
- Sol:-** \* Encapsulation is the concept of bundling data (variables) and methods (functions) that operate on the data into a single unit (class). Voter Slip

- \* It hides the internal details of the object from the outside world.
- \* Encapsulation provides control over how the data is accessed and modified.
- \* How encapsulation helps in organizing and protecting data.

General Election to House of the People, 2024

## Voter Slip

- \* Getter and setters control how data is accessed or updated with the option to add checks.
- \* It improves maintainability by letting you change internal code without affecting the rest of the program.

General Election to House of the People, 2024

## Voter Slip

- Code
- From ..... (No. & Name of AC)
- within the ..... Parliamentary Constituency
- Serial number of elector as per column 1 of the Register of Voters
- Serial number of elector as entered in the electoral roll
- Initials of Polling Officer

# General Election to House of the People, 2024

## Voter Slip

- ⑤ Explain the roles of abstract classes and interfaces in inheritance. Discuss how they provide a blueprint for subclasses and establish a control for methods. provides examples of both abstract classes and interfaces.

**Sol:-** Abstract class:

- \* An abstract class cannot be instantiated.
- \* It is meant to be extended by other classes.
- Voter Slip**
- \* It can have both abstract (no implementation) and concrete (with implementation) methods.
- \* It provides a base for subclasses to inherit shared behaviour.
- \* Subclasses must implement abstract methods but can use concrete ones.

## Interfaces:

# General Election to House of the People, 2024

## Voter Slip

- \* An interface defines a contract with method declarations but no implementations.
- \* Classes implementing an interface must provide implementations for all its methods.
- \* Interfaces support multiple inheritance by allowing a class to implement multiple interfaces.

# General Election to House of the People, 2024

## Voter Slip

- Abstract class Animal
- From ..... (No. & Name of AC)
- within the abstract void sound(); Parliamentary Constituency
- Serial number of elector as per column 1 of the Register of Voters
- Serial number of elector as entered in the electoral roll
- System.out.println("This animal is sleeping");
- Initials of Polling Officer

General Election to House of the People

```
class Dog extends Animal {  
    void sound() {  
        System.out.println("Bark");  
    }  
}
```

### Interfaces :-

```
Interface Animal {  
    void sound();  
}
```

```
Class Dog implements Animal {  
    public void sound() {  
        System.out.println("Bark");  
    }  
}
```

Q) Describe the purpose of the super keyword in Java. Explain how it can be used to access members of the superclass and call superclass constructors.

- Sol:-
- \* The purpose of the super keyword is:-
  - \* The super keyword refers to the immediate superclass of a subclass.
  - \* It is used to access field of the superclass.
  - \* It allows calling methods of the superclass.
  - \* It invokes the constructor of the superclass to initialize it.
  - \* It helps resolve conflicts between subclass and superclass members.

### Explain

- \* Access superclass fields with the same name.
- \* Call superclass methods, even if overridden in the subclass.
- \* Invoke the superclass constructor.
- \* Resolve conflicts between subclass and superclass members.

+ reuse superclass behavior without duplicating code.

### Code

Q) Explain the concept of thread synchronization, and its importance in multithreading.

Sol:-

### concept of Thread Synchronization:-

- \* Ensures that multiple threads can access shared resources in a controlled manner.
- \* Prevents multiple threads from executing critical sections of code simultaneously.
- \* Avoids issues like race conditions and data inconsistency.
- \* Helps maintain program reliability and thread safety.

### Importance

- \* Protects shared resources from simultaneous modification.
- \* Prevents unpredictable behaviour in multithreaded programs.
- \* Ensures data integrity during concurrent thread execution.
- \* Facilitates smooth communication and coordination between threads.

### Code

**Voter Slip**

③ Explain the difference b/w heap and stack memory.

From ..... (No. & Name of AC)

within the **Heap**

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

- \* ~~Stack objects and class instances.~~
- \* Allocated dynamically at run time.

Initials of Polling Officer .....

**Stack**  
Parliamentary Constituency

- \* Shared methods calls, local variables and Parameters.
- \* Allocated automatically when methods are called

\* managed by garbage collection.

General Election to House of the People, 2024

**Voter Slip**

From ..... (No. & Name of AC)

within the .....  
or stack shrink.

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

④ Explain their respective roles in memory management.

Initials of Polling Officer .....

Parliamentary Constituency

- \* Has a fixed size and is faster than heap memory.

- \* It cannot store object directly.

General Election to House of the People, 2024

**Voter Slip**

From ..... (No. & Name of AC)

⑤ Explain how multiple catch blocks can be used to catch and handle specific exception.

within the ..... Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

Responsible code: multiple catch blocks are created and write code.

Initials of Polling Officer .....

\* Multiple catch blocks allows handling different types of exception separately.

General Election to House of the People, 2024

**Voter Slip**

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

\* The catch blocks are evaluated in the order they appear, from top to bottom.

Serial number of elector as entered in the electoral roll

Initials of Polling Officer .....

**Voter Slip**

From ..... (No. & Name of AC)

within the **Order**

Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

\* The first catch block that matches the exception type is executed.

Initials of Polling Officer .....

\* If no catch block matches, the exception is passed to the next higher level.

**Code**

From ..... (No. & Name of AC)

within the **Class multiple catch**

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

by {

Initials of Polling Officer .....

int [A] over = new int [B];  
or [B] = 10;

General Election to House of the People, 2024

**Voter Slip**

From ..... (No. & Name of AC)

within the **System.out.print("Arithmatic error occurred")**

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

System — { some other exception occurred }

Initials of Polling Officer .....

General Election to House of the People, 2024

**Voter Slip**

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

Initials of Polling Officer .....

Q1 Discuss the factory method design pattern. Explain how it allows the creation of objects without specifying the exact class of object to be created.

- Sol:-
- \* The factory method allows object creation without specifying the exact class.
  - \* Subclasses override the factory method to create specific objects.
  - \* It separates object creation from the rest of the code.
  - \* It allows adding new object types without changing existing code.
  - \* Client code interacts with the factory, not specific classes.

The factory method allows object creation by using a common interface or abstract class. The client calls the factory method, and the subclasses decide which class to instantiate - this approach hides the exact class and provides flexibility for future changes.

Q2 Define the concept of design patterns in software engineering. Explain why design patterns are important in development. How they contribute to code reusability and maintainability.

Sol:- Design patterns in software engineering are general, reusable solutions to common problems faced during software design. They are not code templates but abstract guidelines to solve specific design issues.

- \* IMPORTANCE
- \* Provide proven solution, improving efficiency and consistency.
- \* Help design flexible, scalable and maintainable system.
- \* Promote code reusability with tested and optimized pattern.

\* Enable easier system updates by offering adoptable structure.

Q3 What is object? How are they created?

Sol:- Objects are instances of a class that represent real-world entities. They contain fields (data) and methods (behaviour) defined in the class.

Object are created using the new keyword which allocates memory and invokes the constructor.

Example:-

Class car {

String brand;

void drive();

System.out.println(brand + " is driving");

}

Public class main {

Public static void main (String [args]) {

car myCar = new car();

myCar.brand = "Toyota";

~~myCar~~ myCar.drive();

}

Q4 What is constructor? Explain Parameterized constructor.

Sol:- A constructor is a special method used to initialize objects. It is called automatically when an object is created.

\* It has the same name as the class and does not have a return type.

# General Election to House of the People, 2024

## Voter Slip

**A parameterized constructor accepts arguments** (No. & Name of AC)  
**From**  
**to initialize object fields with special** Parliamentary Constituency  
**within the**  
**values during creation to ensure**  
**customized initialization.**

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**Code :-**

Initials of Polling Officer

**class Car {  
String brand;**

# General Election to House of the People, 2024

## Voter Slip

**Car (String b) { brand = b; } .**

**From** (No. & Name of AC)

**void display() { System.out.**

**within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**System.out.println("Brand :" + brand); }**

Initials of Polling Officer

**public static void main(String args) {  
new Car ("Buddy"); }**

# General Election to House of the People, 2024

## Voter Slip

**Car myCar = new Car ("Toyota");  
myCar.display(); }**

**From**

**within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**Q19 Explain the Super keyword with examples**

**Sol:- It is used to call the superclass**

Initials of Polling Officer

**constructor.**

# General Election to House of the People, 2024

## Voter Slip

**In a subclass constructor, super.(name).Com** (No. & Name of AC)

**within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

Initials of Polling Officer

**be used to call the constructor of the superclass and initialize inherited fields.**

# General Election to House of the People, 2024

## Voter Slip

**From** (No. & Name of AC)

**Within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**Class Dog extends Animal {**

Initials of Polling Officer

**Dog (String name);**

**super(name);**

# General Election to House of the People, 2024

## Voter Slip

**From** (No. & Name of AC)

**Within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**public static void main(String args) {  
new Dog ("Buddy"); }**

Initials of Polling Officer

# General Election to House of the People, 2024

## Voter Slip

**(15) How does a class** (No. & Name of AC)

**interfaces into a package?**

**Within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**\* At the top of the Java file, use the package keyword to specify the package name.**

Initials of Polling Officer

**\* Write the class or interface after the package declaration.**

# General Election to House of the People, 2024

## Voter Slip

**From** (No. & Name of AC)

**Within the** Parliamentary Constituency

Serial number of elector as per column 1 of the Register of Voters

Serial number of elector as entered in the electoral roll

**public static void main(String args) {  
HelloWorld h = new HelloWorld(); }**

Initials of Polling Officer

**MyClass myClass = new MyClass();  
obj.greet();**

⑥ what is a finally block ? when and how it is used ? example.

Sol It is used to define code that will always execute, regardless of whether an exception is thrown or not. It is typically used for cleanup operations like closing files.

Uses:-

- After a try block.
- To ensure certain code run after try and catch blocks, even if no exception occur or an exception is caught.

Code:-

public class Finally {

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

```
try {
```

```
System.out.println("In try block");
```

```
int result = 10 / 0;
```

```
catch (ArithmeticException e) {
```

```
System.out.println("In catch block");
```

```
finally {
```

```
System.out.println("In the finally block");
```

⑦ what do you mean by method overloading?

Def

It is defining multiple methods with the same name but different parameters within the same class. It allows you to perform similar operations with different types or number of inputs.

Code

Class overloading :-

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

```
overload ob = new overload();
```

~~ob~~

```
ob.test();
```

```
ob.test(10);
```

}

Class overload

```
{ void test() {
```

```
System.out.println("no parameter");
```

```
} void test(int a) {
```

```
System.out.println("a:" + a);
```

}

Output

no param

10

⑧

Define an exception 'no match exception' and throw it when a string is not equal to 'India'. Write a program that uses try catch exception.

## General Election to House of the People, 2024

### Voter Slip

From ..... (No. & Name of AC)  
 18. What is multithreading? How it's important  
 the performances?  
 within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters  
 Serial number of elector as  
 entered in the electoral roll.

\* It is a program to run multiple  
 threads concurrently, where each  
 instead is a light-weight process.  
 Initials of Polling Officer .....  
 and improve the performance of applications  
 by allowing multiple tasks to be

General Election to House of the People, 2024

### Voter Slip

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

\* If executes tasks in parallel, reducing  
 Initials of Polling Officer ..... overall execution time.

to it keeps application responsive by

General Election to House of the People, 2024

### Voter Slip

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

19. What are command line arguments?

How they are useful?

General Election to House of the People, 2024

### Voter Slip

20. It's values passed to a program  
 From ..... (No. & Name of AC)

Within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

an array of strings.

## General Election to House of the People, 2024

### Voter Slip

From ..... (No. & Name of AC)  
 uses :-

\* They allow user to pass input to a  
 program at runtime without

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

\* They are useful for providing dynamic  
 Initials of Polling Officer ..... values like file paths.

General Election to House of the People, 2024

### Voter Slip

20. What is type casting? Why it's  
 required in programs? (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

\* It is a process of converting a  
 variable from one data type to  
 another. It is required when you need

Initials of Polling Officer ..... to perform operations on different  
 types of data or when available

General Election to House of the People, 2024

need to convert Voter Slip  
 another type. (No. & Name of AC)

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

\* It allows operation different data type.

Initials of Polling Officer ..... to ensure correct data conversion

avoiding unexpected results.

General Election to House of the People, 2024

\* It helps fit values into more suitable  
 data types, optimizing (No. & Name of AC)

From ..... (No. & Name of AC)

within the ..... Parliamentary Constituency

Serial number of elector as per  
 column 1 of the Register of Voters

Serial number of elector as  
 entered in the electoral roll

Initials of Polling Officer .....

Q) Explain the purpose of garbage collection in JVM?

Sol:- Garbage Collection in Java Virtual Machine is the automatic process of reclaiming memory by deleting objects that are no longer in use.

Properties:-

- \* frees up memory used by unreferenced objects.
- \* Prevents memory leaks.
- \* Optimizes program performance by managing memory automatically.

Q) What are FileInputStream and FileOutputStream?

Sol:- FileInputStream

- \* It is used to read data from a file.
- \* Read data byte by byte.
- \* Commonly used for reading binary data. (e.g. image, audio files).

FileOutputStream:-

- \* Used to write data to a file.
- \* Writes data byte by byte.
- \* Commonly used for writing binary data to files.

(23) What is synchronization? ~~What~~ why is it important?

Sol:- Synchronization is a mechanism that ensures only one thread can access a shared resource at a time, preventing data inconsistency and ensuring thread safety in multi-threaded programs.

IMPORTANT :-

- \* Prevents data inconsistency when multiple threads access shared data.
- \* Ensures thread safety by allowing only one thread to access critical resources.
- \* Avoids race conditions that can lead to data inconsistency.
- \* Improves data integrity in concurrent execution scenarios.

Q) What are the desirable properties of a design pattern?

Sol:-

- \* Reusable in different situations.
- \* Scalable as the system grows.
- \* Easy to maintain and update.
- \* Efficient in performances and resource usage.
- \* Flexible for future changes without major modification.