Java Session 3 Notes

Topics Covered

1. Constructors
2. Polymorphism
3. Abstraction
4. Abstract Class
5. Interface

---

1. Constructors in Java

Definition

A constructor is a special method that is automatically called when an object is created. It is used to initialize object variables.

Characteristics of Constructor

1. Constructor name must be same as class name
2. Constructor has no return type (not even void)
3. Constructor is called automatically
4. Constructor is used to initialize variables

Syntax

```
class Student {
    Student() {
        // constructor body
    }
}
```

Types of Constructors

1. Default Constructor

Provided automatically by Java if no constructor is written.

```
class Student {
}
```

---

2. No-Argument Constructor (User-defined)

Created by programmer and does not accept parameters.

```
class Student {
    Student() {
        System.out.println("Student Created");
    }
}
```

---

3. Parameterized Constructor

Accepts parameters and initializes object variables.

```
class Student {
    String name;

    Student(String n) {
        name = n;
    }
}
```

Example Program

```
class Student {
    String name;
    int age;
```

```
    Student(String n, int a) {
        name = n;
        age = a;
    }

    void display() {
        System.out.println(name + " " + age);
    }
}
```

Constructor vs Method

Constructor	Method
Same name as class	Any valid name
No return type	Must have return type
Called automatically	Called manually
Used to initialize object	Used to perform operations

Important Points

- Constructor initializes object
- Constructor name must match class name
- Constructor has no return type
- Constructor is called automatically
- Constructor can be overloaded
- Constructor cannot be inherited
- Constructor is used with new keyword

---

2. Polymorphism in Java

Definition

Polymorphism means one name having multiple behaviors.

Example:
Person can act as Teacher, Student, or Employee.

Types of Polymorphism

1. Compile-Time Polymorphism (Method Overloading)

Same method name but different parameters.

```
class MathOperation {

    int add(int a, int b) {
        return a + b;
    }

    int add(int a, int b, int c) {
        return a + b + c;
    }
}
```

Decided at compile time.

---

2. Run-Time Polymorphism (Method Overriding)

Child class provides its own implementation of parent method.

```
class Animal {
    void sound() {
        System.out.println("Animal makes sound");
    }
}

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

Decided at runtime.

Method Overloading vs Method Overriding

Method Overloading	Method Overriding
Same class	Parent and child class
Different parameters	Same parameters
Compile time	Runtime
Inheritance not required	Inheritance required

Advantages of Polymorphism

- Code reuse
- Flexibility
- Improves readability
- Supports dynamic binding

---

3. Abstraction in Java

Definition

Abstraction is the process of hiding implementation details and showing only essential features to the user.

Real Life Example

ATM Machine

User can:

- Withdraw money
- Check balance

User cannot see internal processing.

Abstraction is achieved using

1. Abstract Class
2. Interface

---

4. Abstract Class

Definition

Abstract class is declared using abstract keyword. It can have abstract methods and normal methods.

Syntax

```
abstract class Animal {
   abstract void sound();
}
```

Example

```
abstract class Animal {

   abstract void sound();

}

class Dog extends Animal {

   void sound() {
```

```
    System.out.println("Dog barks");
  }

}
```

Important Points

- Declared using abstract keyword
- Cannot create object directly
- Can contain abstract and normal methods
- Must be inherited

---

5. Interface in Java

Definition

Interface is a fully abstract class used to achieve abstraction.

Syntax

```
interface Shape {
  void draw();
}
```

Example

```
class Circle implements Shape {

  public void draw() {
    System.out.println("Drawing Circle");
  }

}
```

Keywords Used

- extends is used for abstract class inheritance
- implements is used for interface implementation

---

6. Abstract Class vs Interface

Abstract Class	Interface
Uses abstract keyword	Uses interface keyword
Can have abstract and normal methods	Traditionally only abstract methods
Uses extends	Uses implements
Does not support multiple inheritance	Supports multiple inheritance

---

7. Advantages of Abstraction

- Hides implementation details
- Improves security
- Reduces complexity
- Improves maintainability
- Supports loose coupling

---

8. Memory Understanding

Example from class diagram:

B b1 = new B();

```
b1.setX(10);
b1.setY(20);

b1.displayX();
b1.displayY();
```

Explanation:

- Object is created in heap memory
- Variables are initialized
- Methods access and display values

Important Exam Definitions

Constructor:
A constructor is a special method used to initialize objects and is called automatically when an object is created.

Polymorphism:
Polymorphism means one name having multiple behaviors.

Abstraction:
Abstraction is hiding implementation details and showing only essential features.