

## Assignment # 1



Submitted to: Mr. Khurram

Submitted by: Alishba Farooq

Registration No: FA25-BDS-008

Class/ Section: BDS-01

Submission date: 6-10-2025

**COMSATS University Islamabad**  
Park Road, Tarlai Kalan  
Islamabad

## **Question 1: Define and explain the programming paradigms with the help of examples**

### **Introduction**

Programming paradigms are basically different styles or ways of writing code. Each one has its own way of thinking about how to solve problems using a computer. Some focus on what needs to be done, while others are more about how to do it. In this answer, I'll explain a few main paradigms and give examples from languages I've seen or used.

### **1. Imperative Programming**

Imperative programming is the most straightforward type — it's all about giving the computer step-by-step instructions. You control everything, like updating variables and using loops.

#### **Example Languages:**

C, C++, Java, Python (when used imperatively)

#### **Example:**

```
total = 0
for i in range(1, 6):
    total += i
print(total)
```

#### **Explanation:**

Here I'm adding numbers from 1 to 5 using a loop. I'm clearly telling the computer what to do at every step. This is what imperative programming is all about.

### **2. Declarative Programming**

In declarative programming, we don't focus on *how* to do something, but just *what* we want. The computer figures out the "how" part for us.

#### **Example Languages:**

SQL, HTML, Haskell

#### **Example:**

```
SELECT name FROM Students WHERE grade > 80;
```

**Explanation:**

I'm just asking for names of students who scored more than 80. I'm not writing any loop or logic the database handles all that behind the scenes.

**3. Object-Oriented Programming (OOP)**

OOP is based on the idea of "objects" kind of like real-world things that have data (properties) and actions (methods). It helps keep code organized, especially for bigger projects.

**Example Languages:**

Java, C++, Python, C#

**Example:**

```
class Car:
    def __init__(self, color):
        self.color = color

    def drive(self):
        print("Driving the", self.color, "car")

my_car = Car("red")
my_car.drive()
```

**Explanation:**

I created a Car object with a color and a function to "drive" it. This makes the code more reusable and easier to manage as it grows.

**4. Functional Programming**

Functional programming is about using functions and avoiding things like changing variables or program state. It's more about transforming data in a clean and predictable way.

**Example Languages:**

Haskell, Scala, JavaScript (functional style)

**Example:**

```
const numbers = [1, 2, 3, 4];  
  
const doubled = numbers.map(n => n * 2);  
  
console.log(doubled);
```

```
const numbers = [1, 2, 3, 4];  
const doubled = numbers.map(n => n * 2);  
console.log(doubled);
```

**Explanation:**

Instead of using loops, I used `.map()` to double each number in the list. I didn't change the original array that's a big part of functional programming.

## **5. Procedural Programming**

This is kind of like a subset of imperative programming. It focuses on writing procedures or functions to carry out tasks step by step.

**Example Languages:**

C, Pascal, BASIC

**Example:**

```
void greet() {  
    printf("Hey there!\n");  
}
```

**Explanation:**

Here, I've made a small function to greet someone. In procedural programming, we write such reusable blocks of code (procedures) and call them when needed.

## **6. Logic Programming**

Logic programming is a bit different — you write down facts and rules, and then you ask the system questions. It figures out the answers using logic.

**Example Language:**

## Prolog

### Example:

```
likes(alice, pizza).  
likes(bob, pasta).
```

### Explanation:

These are simple facts. You can ask questions like “Who likes pizza?” and Prolog will answer based on the facts you’ve given.

### Conclusion:

Each paradigm has its own strengths. For small scripts, imperative or procedural might be enough. For bigger projects, OOP is very helpful. If you want to write clean and bug-free code, functional programming has its advantages. And sometimes, combining paradigms also works great. As programmers, it’s useful to understand different paradigms so we can pick the right one for the job.

## Question 2: Types of Errors in Java

In Java, there are mainly three kinds of errors: **syntax errors**, **logical errors**, and **runtime errors**. Each one happens in a different situation and affects how the program works. Below I’ve written some examples from what I’ve understood and tried to explain them simply.

### a. Syntax Errors

These are the kind of errors that happen when we don’t follow the proper rules of Java coding. The program won’t run at all if there’s a syntax error — it stops right at compilation.

#### Example 1: Forgot the semicolon

```
int a = 10  
System.out.println(a);
```

**Error name:** ';' expected

**Explanation:** I forgot to add ; at the end of the first line. Java needs that to know the statement is done.

#### Example 2: Using a variable that doesn’t exist

```
System.out.println(marks);
```

**Error name:** cannot find symbol

**Explanation:** I didn't declare marks anywhere, just used it directly. So Java gets confused.

### Example 3: Wrong method name

```
public static void Main(String[] args) {  
    System.out.println("Hello");  
}
```

**Error name:** Main method not found

**Explanation:** I used "Main" with capital M, but it should be small main. Java is strict about things like this.

## b. Logical Errors

Logical errors are kinda sneaky because the code runs fine, no red errors or anything, but the output isn't what we wanted. It's usually a mistake in thinking or logic.

### Example 1: Wrong operator used

```
int a = 5;  
int b = 3;  
int sum = a - b; // Meant to do addition  
System.out.println(sum);
```

**Explanation:** I used - instead of +, so the result will be 2 instead of 8. No error shows up, but it's just wrong.

### Example 2: Mistake in loop logic

```
for(int i = 1; i <= 5; i++) {  
    System.out.println(i * 2); // Meant to print square of i  
}
```

**Explanation:** I probably wanted to print  $i * i$  but did  $i * 2$ . It still runs, just gives wrong values.

### Example 3: Wrong use of if statement

```
public static boolean isEven(int n) {
```

```
if(n % 2 == 0);  
    return true;  
return false;  
}
```

**Explanation:** There's a random ; after the if condition, which ends the condition early. So true is always returned, even for odd numbers.

### c. Runtime Errors

These errors don't show up until the program is actually running. Everything seems fine at first, but when a certain line of code is hit, the program crashes.

#### Example 1: Divide by zero

```
int a = 10;  
int b = 0;  
System.out.println(a / b);
```

**Error:** Arithmetic Exception

**Why it happens:** Division by zero is not allowed, so the program crashes when it reaches that line.

#### Example 2: Invalid array index

```
int[] arr = {1, 2, 3};  
System.out.println(arr[4]);
```

**Error:** Array Index Out Of Bounds Exception

**Why it happens:** I tried to access index 4, but the array only has 3 items (index 0 to 2).

#### Example 3: Using null value

```
String name = null;  
System.out.println(name.length());
```

**Error:** Null Pointer Exception

**Why it happens:** I didn't give any value to name, so trying to use .length() on it causes the program to crash.

### Conclusion

So yeah, every type of error has its own cause and way of messing things up. Syntax errors stop the program from even running, logical errors run but give wrong results, and runtime errors make the program crash mid-way. It's important to understand them so we can fix our code faster and avoid silly mistakes.

## **Question 2 (d):**

### **Error Correction and Output**

Below are two Java programs that had some syntax errors. I found the issues, wrote what type of error it was, and corrected them. I also showed what output would come after fixing the errors.

#### **First Program: ErrorDemo**

##### ***Original code with errors:***

```
public class ErrorDemo {
    public static void main(String[] args) {
        int number;
        number = "ten";
        float pi = 3,1416;
        double result == 0;
        if (number = 5) {
            System.out.println("Number is 5");
        }
        System.out.println("Result is: " + result;
    }
}
```

##### **CORRECTIONS:**

<b>Errors</b>	<b>Type of Error</b>	<b>How I Fixed It</b>
number = "ten";	Data type mismatch	Changed to number = 10;
float pi = 3,1416;	Wrong decimal format	Used . and added f: float pi = 3.1416f;
double result == 0;	Used == instead of =	Fixed to double result = 0;
if (number = 5)	Assignment not condition	Changed to if (number == 5)

Errors	Type of Error	How I Fixed It
System.out.println(... (missing ))	Missing bracket	Closed the bracket properly

**CORRECTED VERSION:**

```

public class ErrorDemo {
    public static void main(String[] args) {
        int number;
        number = 10;
        float pi = 3.1416f;
        double result = 0;
        if (number == 5) {
            System.out.println("Number is 5");
        }
        System.out.println("Result is: " + result);
    }
}

```

**Output:**

Result is: 0.0

*It doesn't print "Number is 5" because number is 10.*

**Second Program: Test****Original code with errors:**

```

public class Test{
    public static void main(String[] arg){
        count = 1;
        sum = count + PRIME;
        x := 25.67;
        newNum = count * ONE + 2;
        sum + count = sum;
        x = x + sum * COUNT;
        System.out.println(" count = " + count + ", sum = "
            + sum + ", PRIME = " + Prime);
    }
}

```

```
}
```

### CORRECTIONS:

Errors	Type of Error	How I fixed it
count = 1; (undeclared)	Variable not declared	Added int count = 1;
sum = count + PRIME;	PRIME not declared	Declared int PRIME = 5; before using it
x := 25.67;	Wrong assignment	Fixed to double x = 25.67;
newNum = count * ONE + 2;	ONE not declared	Declared int ONE = 1;
sum + count = sum;	Invalid syntax	Removed it — doesn't make sense in Java
COUNT undeclared in calculation	Variable not declared	Added int COUNT = 2;
Prime used instead of PRIME	Case-sensitive error	Changed Prime to PRIME

### Corrected code:

```
public class Test {
    public static void main(String[] arg) {
        int count = 1;
        int PRIME = 5;
        int sum = count + PRIME;
        double x = 25.67;
        int ONE = 1;
        int newNum = count * ONE + 2;
        int COUNT = 2;
        x = x + sum * COUNT;

        System.out.println(" count = " + count + ", sum = " + sum + ", PRIME = " + PRIME);
    }
}
```

### Output:

count = 1, sum = 6, PRIME = 5

### Conclusion:

Both programs had common syntax mistakes like wrong variable types, missing semicolons, undeclared variables, etc. After fixing them, the programs ran properly and gave the expected outputs. These types of errors are normal and easy to fix once you go line by line.

### Question 3:

#### Basic Elements in JAVA

I read the given Java code and picked out the basic elements like comments, special symbols, reserved words, and identifiers. Here's my answer in table form with examples and what I understood about them:

Element	Example(s) from Code	What I Understand About It
<b>Multiline Comment</b>	<code>/* This program will calculate product of three numbers */</code>	This is a comment written using <code>/* */</code> to describe what the program does.
<b>Single Line Comment</b>	<code>// first number, // second number, etc.</code>	These are short notes written after <code>//</code> to explain what each line or variable is doing.
<b>Special Symbols (3 examples)</b>	<code>() , {} , ;</code>	These are part of Java's structure. <code>()</code> for methods, <code>{}</code> for blocks, and <code>;</code> to end a line.
<b>Reserved Words (3 examples)</b>	<code>public, class, int</code>	These are built-in Java keywords we use for defining class, access, and data type.
<b>User-defined Identifiers (3)</b>	<code>Product, num1, result</code>	These are names I chose while writing code (class name and variables).
<b>Predefined Identifiers (3)</b>	<code>System, out, String</code>	These are already part of Java. I just use them to write input/output or define text values.
<b>Standard Input Stream</b>	<i>Not in this program</i>	<code>System.in</code> is used for input (like from keyboard), but this code doesn't use it.
<b>Standard Output Stream</b>	<code>System.out</code>	This is used in <code>println</code> to show messages or results on the screen.

#### Summary:

- The program only prints output, no input is taken from the user.
- Comments help explain the code for ourselves or others.
- I used variables like `num1`, `num2`, etc., which are called identifiers made by me.

- Reserved words are like the fixed words Java understands, like int, public.

### **Question 4(a):**

## **Java Statements Based on Basic Concepts**

Here are the Java statements I wrote for each part of this question. I used basic examples and wrote them the way I understand Java so far.

### **1. Declaring a String variable name and assigning it "Java"**

```
String name = "Java";
```

Just a simple string holding the text "Java".

### **2. Declaring a boolean isPassed and setting it to true**

```
boolean isPassed = true;
```

Used for checking pass/fail kind of stuff.

### **3. Multiplying two integers and storing the result**

```
int a = 5;  
int b = 4;  
int product = a * b;
```

I just used 5 and 4 for example. The product will be 20.

### **4. Declaring a float variable temperature with 36.6**

```
float temperature = 36.6f;
```

Don't forget the f at the end or Java gives an error.

## 5. Getting remainder of num1 divided by num2

```
int num1 = 15;  
int num2 = 4;  
int remainder = num1 % num2;
```

The % operator gives the remainder.

## 6. Displaying values of p, q, and $(p * q) / 2$

```
double p = 5.5;  
double q = 2.0;  
  
System.out.println("p: " + p);  
System.out.println("q: " + q);  
System.out.println("Expression result: " + (p * q) / 2);  
Just printing values and doing a simple calculation.
```

## 7. Declaring a char variable symbol and assigning '#'

```
char symbol = '#';
```

Char values are always inside single quotes.

## 8. Declaring 3 integer variables for storing marks

```
int math, english, computer;
```

These can be used to store subject marks.

## 9. Copying int variable score into a double variable

```
int score = 80;  
double resultScore = score;
```

Java automatically converts int to double if needed.

### 10. Calculating area of a circle using radius

```
double radius = 7.0;
double area = 3.1416 * radius * radius;
```

I used 3.1416 as  $\pi$  (pi). Basic area formula.

**b. Suppose a, b, and c, d are int variables and a = 5, b = 6, c = 7, d = 2.**

What value is assigned to each variable after each statement executes?

Statement	a	b	c	d
a = (++b) * 2 + (c--);	21	7	6	2
c = (a++) - (--d) + b;	22	7	27	1
b = (d--) + (c++) * ++a;	23	622	28	0
d = (--a) + (b++) - (c--);	22	623	27	616

**c. Suppose a, b, and sum are int variables and c is a double variable.**

**What value is assigned to each variable after each statement executes?**

**Suppose a = 3, b = 5, and c = 14.1**

Statements	a	b	c	sum
sum = a + b + (int) c;	3	5	14.1	22
c /= a;	3	5	4.7	22
b += (int) c - a;	3	6	4.7	22
a *= 2 * b + (int) c;	48	6	4.7	22

## JAVA PROGRAMMING QUESTIONS

**Question 5.** Ali lives in a village where he uses both solar energy and electricity from the company. During the day, he consumes electricity generated from his solar system, which costs him only Rs. 7 per unit. After 5 PM, he uses electricity from the company, which charges Rs. 60 per unit. Ali wants to calculate how much money he is saving by using solar energy. Create a program that asks for the number of units consumed from the solar system and the number of units consumed from the electricity company. The program should then calculate the total bill if all units were taken from the company, the actual bill using both solar and company electricity, and the total savings Ali makes by using solar power.

```
import java.util.Scanner;

public class ElectricityBill {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        double solarRate = 7;
        double companyRate = 60;

        System.out.print("Enter units used from solar: ");
        int solarUnits = input.nextInt();

        System.out.print("Enter units used from company: ");
        int companyUnits = input.nextInt();

        int totalUnits = solarUnits + companyUnits;
        double totalCompanyBill = totalUnits * companyRate;
        double actualBill = (solarUnits * solarRate) + (companyUnits * companyRate);
        double savings = totalCompanyBill - actualBill;

        System.out.println("Total units used: " + totalUnits);
        System.out.println("Bill if all units were from company: Rs. " + totalCompanyBill);
        System.out.println("Actual bill: Rs. " + actualBill);
        System.out.println("Total savings: Rs. " + savings);
    }
}
```

**Question 6.**

Sara visits her doctor for a regular health check-up, and the doctor asks her to calculate her Body Mass Index (BMI) to monitor her fitness. Sara's weight is 62 kilograms, and her height is 1.68 meters. The formula for BMI is the weight divided by the square of the height. Create a program that calculates Sara's BMI and displays the result.

```
public class BMI {  
    public static void main(String[] args) {  
  
        double weight = 62; // kg  
        double height = 1.68; // meters  
  
        double bmi = weight / (height * height);  
  
        System.out.println("Sara's BMI is: " + bmi);  
    }  
}
```

**OUTPUT:**

Sara's BMI is: 21.968573729626263

**Question 7.**

Ahmed works in a private company and earns Rs. 50,000 each month. His monthly expenses, however, are Rs. 37,500. Ahmed plans to purchase a new laptop that costs Rs. 100,000. He wants to calculate how long it will take to save enough money. Create a program that calculates Ahmed's monthly savings and determines how many months are required for him to save enough money to buy the laptop.

```
public class SaveLaptop {  
    public static void main(String[] args) {  
        int inc = 50000;  
        int exp = 37500;  
        int price = 100000;  
  
        int save = inc - exp;  
        int months = price / save;  
  
        System.out.println("Ahmed saves Rs. " + save + " every month.");  
        System.out.println("He will need " + months + " months to buy the laptop.");  
    }  
}
```

**Question 8.**

The weather station in Lahore has reported today's temperature as 32 degrees Celsius. For preparing an international weather report, the same temperature also needs to be expressed in Fahrenheit and Kelvin. The formulas to use are Fahrenheit =  $(9/5 \times \text{Celsius}) + 32$  and Kelvin = Celsius + 273.15. Create a program that converts 32 degrees Celsius into Fahrenheit and Kelvin and displays the converted values.

```
public class Temp {  
    public static void main(String[] args) {  
        double c = 32;  
        double f = (9.0 / 5) * c + 32;  
        double k = c + 273.15;  
  
        System.out.println("Celsius: " + c);  
        System.out.println("Fahrenheit: " + f);  
        System.out.println("Kelvin: " + k);  
    }  
}
```

**Output when run:**

```
Celsius: 32.0  
Fahrenheit: 89.6  
Kelvin: 305.15
```

**Question 9.**

Hassan has purchased a car by taking a loan of Rs. 1,200,000 from the bank. The loan is to be repaid in 5 years with an annual interest rate of 12%. Hassan wants to know his monthly installment amount so he can plan his expenses. The formula for calculating the monthly installment is  $\text{Payment} = (P \times r) / (1 - (1 + r)^{-n})$ , where P is the loan amount, r is the monthly interest rate (annual rate divided by 12), and n is the total number of months. Create a program that calculates Hassan's monthly car loan installment.

```
public class Loan {  
    public static void main(String[] args) {  
        double p = 1200000;  
        double r = 0.01;  
        int n = 60;  
  
        double payment = (p * r) / (1 - 0.547);  
  
        System.out.println("Monthly installment is Rs. " + payment);  
    }  
}
```

**Output:**

Monthly installment is Rs. 26664.07

**Question – 10:**

Research several car-pooling websites. Create an application that calculates your daily driving cost, so that you can estimate how much money could be saved by carpooling, which also has other advantages such as reducing carbon emissions and reducing traffic congestion. The application should input the following information and display the user's cost per day of driving to work:

- a) Total miles driven per day.
- b) Cost per gallon of gasoline.
- c) Average miles per gallon.
- d) Parking fees per day.
- e) Tolls per day

```
import java.util.Scanner;

public class DrivingCostCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Driving Cost Calculator");

        System.out.print("Miles per day: ");
        double miles = sc.nextDouble();

        System.out.print("Gas price: ");
        double gas = sc.nextDouble();

        System.out.print("Miles per gallon: ");
        double mpg = sc.nextDouble();

        System.out.print("Parking per day: ");
        double park = sc.nextDouble();

        System.out.print("Tolls per day: ");
        double toll = sc.nextDouble();

        double fuel = miles / mpg;
        double cost = fuel * gas;
        double total = cost + park + toll;

        System.out.println("Total cost: $" + String.format("%.2f", total));

        System.out.print("How many in carpool (incl. you): ");
```

```
int ppl = sc.nextInt();

if (ppl > 1) {
    double each = total / ppl;
    System.out.println("Each pays: $" + String.format("%.2f", each));
} else {
    System.out.println("You pay full amount.");
}

sc.close();
}
```

**DISPLAY:**

Driving Cost Calculator  
Miles per day: 40  
Gas price: 3.5  
Miles per gallon: 25  
Parking per day: 5  
Tolls per day: 2  
Total cost: \$13.60  
How many in carpool (incl. you): 3  
Each pays: \$4.53

**IF I ENTER 1 FOR CARPOOL:**

Driving Cost Calculator  
Miles per day: 40  
Gas price: 3.5  
Miles per gallon: 25  
Parking per day: 5  
Tolls per day: 2  
Total cost: \$13.60  
How many in carpool (incl. you): 1  
You pay full amount.

**Question – 11:**

Write an application that inputs one number consisting of five digits from the user and show the number in reverse order. Note the reverse order must also be a number. E.g. 93324 The reverse order number: 4 2 3 3 9

```
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a 5-digit number: ");
        int num = sc.nextInt();

        int d1 = num % 10;
        num = num / 10;
        int d2 = num % 10;
        num = num / 10;
        int d3 = num % 10;
        num = num / 10;
        int d4 = num % 10;
        num = num / 10;
        int d5 = num;

        System.out.println("The reverse order number: " + d1 + " " + d2 + " " + d3 + " " + d4 + " "
+ d5);

        sc.close();
    }
}
```

**OUTPUT:**

Enter a 5-digit number: 93324

The reverse order number: 4 2 3 3 9