Java Fundamentals:

Task 1: Data Types/Variables

Write a program that declares two integer variables, swaps their values without using a third variable, and prints the result.

```
package com.wipro.fundamental;
public class Swap
 {
      public static void main(String[] args)
    {
        int a = 5;
        int b = 10;

        System.out.println("Before swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);

         a = a + b
         b = a - b;
         a = a - b;
        System.out.println("After swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
      }
}
```

Task 2: Operators

Create a program that simulates a simple calculator using command-line arguments to perform and print the result of addition, subtraction, multiplication, and division..

```
package com.wipro.fundamental;

public class Calculator
{
       if (args.length != 3) {
         System.out.println("Usage: java SimpleCalculator <num1> <operation><num2>");
         System.out.println("Operations: +, -, *, /");
         return;
     }
     double num1 = Double.parseDouble(args[0]);
     String operation = args[1];
     double num2 = Double.parseDouble(args[2]);
     double result = 0;

     switch (operation) {
         case "+":
             result = num1 + num2;
             break;
         case "-":
             result = num1 - num2;
             break;
         case "*":
             result = num1 * num2;
             break;
         case "/":
             if (num2 == 0) {
                 System.out.println("Error: Division by zero is not allowed.");
                 return;
             }
             result = num1 / num2;
             break;
         default:
             System.out.println("Error: Invalid operation. Use +, -, *, or /.");
             return;
     }

     System.out.println("Result: " + result);
}
```

Task 3: Control Flow

Write a Java program that reads an integer and prints whether it is a prime number using a for loop and if statements.

```
package com.wipro.fundamental;
import java.util.Scanner;
public class PrimeNumber
{
        public static void main(String[] args)
        {
                        Scanner scanner = new Scanner(System.in);
                        System.out.print("Enter an integer: ");
                        int number = scanner.nextInt();
                        scanner.close();

                        boolean isPrime = true;

                        if (number <= 1) {
                            isPrime = false;
                        } else {
                            for (int i = 2; i <= Math.sqrt(number); i++) {
                                if (number % i == 0) {
                                    isPrime = false;
                                    break;
                                }
                            }
                        }
                        if (isPrime) {
                            System.out.println(number + " is a prime number.");
                        } else {
                            System.out.println(number + " is not a prime number.");
                        }
```

```
                 }
             }
```

Task 4: Constructors

Implement a Matrix class that has a constructor which initializes the dimensions of a matrix and a method to fill the matrix with values.

```
package com.wipro.fundamental;
import java.util.Scanner;

      public class Matrix {
          private int rows;
          private int cols;
          private int[][] data;

          // Constructor to initialize the dimensions of the matrix
          public Matrix(int rows, int cols) {
              this.rows = rows;
              this.cols = cols;
              data = new int[rows][cols];
          }
          // Method to fill the matrix with values
          public void fillMatrix() {
              Scanner scanner = new Scanner(System.in);
              System.out.println("Enter the elements of the matrix:");

              for (int i = 0; i < rows; i++) {
                  for (int j = 0; j < cols; j++) {
                      System.out.print("Element at (" + i + ", " + j + "): ");
                      data[i][j] = scanner.nextInt();
                  }
              }

              scanner.close();
          }

          // Method to display the matrix
          public void displayMatrix() {
              System.out.println("Matrix:");
              for (int i = 0; i < rows; i++) {
                  for (int j = 0; j < cols; j++) {
                      System.out.print(data[i][j] + " ");
                  }
                  System.out.println();
              }
          }

          // Main method to test the Matrix class
          public static void main(String[] args) {
              Scanner scanner = new Scanner(System.in);

              // Input dimensions of the matrix
              System.out.print("Enter the number of rows: ");
              int rows = scanner.nextInt();
              System.out.print("Enter the number of columns: ");
              int cols = scanner.nextInt();

              // Create a matrix with the specified dimensions
              Matrix matrix = new Matrix(rows, cols);
              matrix.fillMatrix();
              matrix.displayMatrix();
              scanner.close();
          }
      }
```

Task 5: Inheritance

Create a Shape class with a method area() and extend it with Circle and Rectangle classes overriding the area() method appropriately.

Shape.Java

```
package com.wipro.fundamental;
public abstract class Shape
 {
    public abstract double area();
}
```

Circle.java

```
package com.wipro.fundamental;
public class Circle extends Shape
 {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double area() {
        return Math.PI * radius * radius;
    }
}
```

Rectangle.java

```
package com.wipro.fundamental;

public class Rectangle extends Shape
 {
    private double width;
    private double height;
    // Constructor to initialize width and height
    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    // Override the area method to calculate the area of a rectangle
    @Override
    public double area() {
        return width * height;
    }
}
```

Main.java

```
package com.wipro.fundamental;

      public class Main {
          public static void main(String[] args)
 {
              // Create a Circle object and display its area
              Shape circle = new Circle(5);
              System.out.println("Area of Circle " + circle.area());

              // Create a Rectangle object and display its area
              Shape rectangle = new Rectangle(4, 6);
              System.out.println("Area of Rectangle: " + rectangle.area());
          }
      }
```

Task 6: Packages/Classpath

Create a package com.math.operations and include classes for various arithmetic operations. Demonstrate how to compile and run these using the classpath.

```
project

|

└───src

   └───com

      └───math

         └───operations

            |  Addition.java

            |  Sub.java

            |  Multi.java

            |  Div.java
```

Addition.java

```
 package com.math.operations;

public class Addition {
    public static int add(int a, int b)
{
        return a + b;
    }
}
```

Subtraction.java

```
package com.math.operations;

public class Sub
{
    public static int subtract(int a, int b)
    {
        return a - b;
    }
 }
```

Multiplication.java

```
package com.math.operations;

public class Multi
{
    public static int multiply(int a, int b)
  {
        return a * b;
    }
}
```

Division.java

```
package com.math.operations;

public class Div
 {
    public static double divide(double a, double b)
     {
        if (b == 0) {
            throw new IllegalArgumentException("Division by zero is not allowed.");
        }
        return a / b;
    }
}
```

Main.java

```
package com.math.operations;
import com.math.operations.Addition;
import com.math.operations.Subt;
import com.math.operations.Multi;
import com.math.operations.Div;
public class Main
 {
    public static void main(String[] args)
     {
        int a = 10;
        int b = 5;

        // Test addition
        System.out.println("Addition:" + a + " + " + b + " = " + Addition.add(a, b));

        // Test subtraction
       System.out.println("Sub:" + a + "-" + b + "=" +        Subt.subtract(a,b));

        // Test multiplication
        System.out.println("Multi:"+ a + "*" + b + "=" + Multi.multiply(a, b));

        // Test division
        System.out.println("Div:"+ a + "/" + b + "=" + Div.divide(a, b));
    }
}
```

Task 7: Basic Exception Handling

Write a program that attempts to divide by zero, catches the ArithmeticException, and provides a custom error message.

```
package com.wipro.fundamental;

      public class DivideByZeroExample {
          public static void main(String[] args) {
              try {
                  // Attempt to divide by zero
                  int result = divideByZero(10,0);
                  System.out.println("Result: " + result);
              } catch (ArithmeticException e) {
                  // Catch ArithmeticException and provide custom error message
                  System.out.println("Error: Division by zero is not allowed.");
              }
          }

          public static int divideByZero(int numerator, int denominator) {
              // Attempt division
              return numerator / denominator;
          }
      }
```

```
Error: Division by zero is not allowed.
```