

No	Script	Hasil Running
1	<pre> import java.util.Scanner; abstract class AbstractClass { public abstract void info(); } class CustomClass extends AbstractClass { @Override public void info() { Scanner scanner = new Scanner(System.in); System.out.println("Masukkan informasi: "); String info = scanner.nextLine(); System.out.println("Informasi yang dimasukkan: " + info); } } public class P8 { public static void main(String[] args) { CustomClass customObj = new CustomClass(); customObj.info(); } } </pre>	Masukkan informasi: P8 Informasi yang dimasukkan: P8
2	<pre> import java.util.Scanner; abstract class AbstractShape { public abstract int calculateArea(int length, int width); } class Rectangle extends AbstractShape { @Override public int calculateArea(int length, int width) { return length * width; } } public class P8 { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.println("Masukkan panjang: "); </pre>	Masukkan panjang: 2 Masukkan lebar: 3 Luas persegi panjang: 6

	<pre> int length = scanner.nextInt(); System.out.println("Masukkan lebar: "); int width = scanner.nextInt(); AbstractShape rectangle = new Rectangle(); int area = rectangle.calculateArea(length, width); System.out.println("Luas persegi panjang: " + area); } } </pre>	
3	<pre> import java.util.Scanner; abstract class AbstractLoop { public abstract void repeat(int count); } @Override public void repeat(int count) { for (int i = 0; i < count; i++) { System.out.println("Kamu adalah yang terbaik !!"); } } public class P8 { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.println("Masukkan jumlah perulangan: "); int count = scanner.nextInt(); AbstractLoop repeater = new RepeatApplication(); repeater.repeat(count); } } </pre>	<pre> Masukkan jumlah perulangan: 2 Kamu adalah yang terbaik !! Kamu adalah yang terbaik !! </pre>

4	<pre> import java.util.Scanner; abstract class AbstractChoice { public abstract void makeChoice(int choice); } class ChoiceApplication extends AbstractChoice { @Override public void makeChoice(int choice) { switch (choice) { case 1: System.out.println("Anda memilih 1"); break; case 2: System.out.println("Anda memilih 2"); break; default: System.out.println("Pilihan tidak valid"); } } } public class P8 { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.println("Masukkan pilihan (1 atau 2): "); int choice = scanner.nextInt(); AbstractChoice chooser = new ChoiceApplication(); chooser.makeChoice(choice); } } </pre>	<p>Masukkan pilihan (1 atau 2):</p> <p>1</p> <p>Anda memilih 1</p>
5	<pre> public interface Laporan { void tampilkanLaporan(); } public class ImplementasiLaporan implements Laporan { @Override public void tampilkanLaporan() { System.out.println("Memberi laporan..."); } public static void main(String[] args) { ImplementasiLaporan implementasi = new ImplementasiLaporan(); implementasi.tampilkanLaporan(); } } </pre>	<p>Memberi laporan...</p>
6	<pre> public interface TampilkanData { void tampilkanData(); } </pre>	<p>Berikut adalah datanya...</p> <p>Mencetak data...</p>

	<pre> } public interface CetakData { void cetakData(); } public class Program implements TampilkanData, CetakData { @Override public void tampilkanData() { System.out.println("Berikut adalah datanya..."); } @Override public void cetakData() { System.out.println("Mencetak data..."); } public static void main(String[] args) { Program program = new Program(); program.tampilkanData(); program.cetakData(); } } </pre>	
7	<pre> interface CetakLaporan { void cetakA4(); void cetakA3(); } interface TampilLaporan { void tampilWeb(); void tampilMobile(); } interface Laporan extends CetakLaporan, TampilLaporan { void info(); } class NewClass implements Laporan { @Override public void cetakA4() { System.out.println("Cetak laporan A4"); } @Override public void cetakA3() { System.out.println("Cetak laporan A3"); } @Override public void tampilWeb() { </pre>	<pre> Cetak laporan A4 Cetak laporan A3 Tampil di monitor Tampil di mobile Info laporan: dua ribu </pre>

	<pre> System.out.println("Tampil di monitor"); } @Override public void tampilMobile() { System.out.println("Tampil di mobile"); } @Override public void info() { System.out.println("Info laporan: dua ribu"); } } class LaporanTahunan extends NewClass { public static void main(String[] args) { LaporanTahunan laporanTahunan = new LaporanTahunan(); laporanTahunan.cetakA4(); laporanTahunan.cetakA3(); laporanTahunan.tampilWeb(); laporanTahunan.tampilMobile(); laporanTahunan.info(); } } </pre>	
8	<pre> public interface TampilLaporan { void tampilWeb(); void tampilMobile(); } public interface CetakLaporan { void cetakA4(); void cetakA3(); } public class Program implements TampilLaporan, CetakLaporan { @Override public void tampilWeb() { System.out.println("Tampil di web"); } @Override public void tampilMobile() { System.out.println("Tampil di mobile"); } @Override public void cetakA4() { System.out.println("Cetak laporan A4"); } @Override </pre>	<pre> Tampil di web Tampil di mobile Cetak laporan A3 Cetak laporan A4 Tampil di web Tampil di mobile Cetak laporan A3 Cetak laporan A4 </pre>

	<pre>public void cetakA3() { System.out.println("Cetak laporan A3"); } public static void main(String[] args) { Program objekA = new Program(); objekA.tampilWeb(); objekA.tampilMobile(); objekA.cetakA3(); objekA.cetakA4(); Program objekB = new Program(); objekB.tampilWeb(); objekB.tampilMobile(); objekB.cetakA3(); objekB.cetakA4(); } }</pre>	
--	---	--