

## EXAMPLE QUESTIONS

**Question 1.** What will be the output of the program?

<pre>public class Parent {      public void method1(){         System.out.print("Parent 1 ");     }     public void method2(){         System.out.print("Parent 2 ");     } }</pre>	<pre>public class Child extends Parent {      public Child(){         System.out.println("Child ");     }     public void method1(){         System.out.print("Child 1 ");     }     public void method2(String p){         System.out.print("Child p ");     } }</pre>
<pre>public class GrandChild extends Child {      public void method2(){         System.out.print("Grandchild 2 ");     } }</pre>	<pre>public class PolyTest {     public static void main(String[] args) {         Parent[] children = new Parent[3];         children[0] = new Child();         children[1] = new Parent();         children[2] = new GrandChild();          for(Parent prnt : children){             prnt.method1();             prnt.method2();             System.out.println();         }     } }</pre>

**Answer:**

Child  
Child  
Child 1 Parent 2  
Parent 1 Parent 2  
Child 1 Grandchild 2

**Question 2.** What will be the output of the program?

<pre>public class Parent {      static int x=1;     public Parent() { x++; }     public void method(){         System.out.print(x);     } }</pre>	<pre>public class Child extends Parent {      public Child(){ x++; } }</pre>
<pre>public class GrandChild extends Child {      public GrandChild() { x++; }     public void method(){         System.out.print(x);     } }</pre>	<pre>public class PolyTest {      public static void main(String[] args) {         Parent[] children = new Parent[3];         children[0] = new Child();         children[1] = new Parent();         children[2] = new GrandChild();          for(Parent prnt : children){             prnt.method();             System.out.println();         }     } }</pre>

**Answer:**

7  
7  
7

**Question 3.** What will be the output of the program?

<pre>public class Parent {      public Parent() {         System.out.println("Parent");     }     public Parent(String str){         System.out.print(str);     } }</pre>	<pre>public class Child extends Parent {     public Child(String str){         super(str);     } }</pre>
<pre>public class GrandChild extends Child {     public GrandChild() {         super("Grand");         System.out.println("GrandChild");     } }</pre>	<pre>public class PolyTest {     public static void main(String[] args) {         Parent prnt = new Parent();         Parent child = new Child("Grand");         GrandChild gchild= new GrandChild();     } }</pre>

**Answer:**

Parent  
GrandGrandChild

**Soru 4.** According to the Java code given below,

<pre> class A {     public void method1() {         System.out.print("A1 ");     } }  class B extends A {     public void method2() {         System.out.print("B2 ");     } } </pre>	<pre> class C extends B {     public void method2() {         super.method1();         System.out.print("C2 ");     } }  class D extends C {     public void method1() {         System.out.print("D1 ");     }     public void method2() {         System.out.print("D2 ");         super.method2();     } } </pre>
---	--

Considering the variable definitions given below, write down the outputs of each of the Java method calls to the nearby blank areas.

```

A x = new D();
C y = new D();
A z = new B();
Object o = new C();

```

x.method1();	D1
y.method1();	D1
y.method2();	D2 A1 C2
z.method1();	A1
z.method2();	methods() is undefined
((B) x).method2();	D2 A1 C2
((B) ((C) o)).method1();	A1
((C) z).method1();	ClassCastException