

A

NOME, COGNOME	
NUMERO DI MATRICOLA	

Istruzioni: leggere il codice dei test sui fogli allegati.

Indicare la risposta sul presente foglio, cerchiando la voce A, B o C. Se si prevede un errore indicare la riga e riportare la motivazione nel campo libero. Se si prevede una corretta esecuzione del codice riportarne l'output nel campo libero.

TEST 1	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 2	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 3	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 4		il codice esegue correttamente, e l'output è →	
--------	--	--	--

TEST 5		il codice esegue correttamente, e l'output è →	
--------	--	--	--

TEST 6		il codice esegue correttamente, e l'output è →	
--------	--	--	--

TEST7		il codice esegue correttamente, e l'output è →	
-------	--	--	--

TEST 8		il codice esegue correttamente, e l'output è →	
--------	--	--	--

TEST 9	Riportare la sequenza di V e F								
--------	--------------------------------	--	--	--	--	--	--	--	--

A

Test 1 java A

00	<code>class B{</code>
01	<code> B(){ System.out.print("C");}</code>
02	<code> void f() { System.out.print("D");}</code>
03	<code>}</code>
04	<code>public class A extends B {</code>
05	<code> A(){ System.out.print("A");}</code>
06	<code> void f() { System.out.print("B");}</code>
07	<code> public static void main(String[] args) {</code>
08	<code> A b=new A();</code>
09	<code> B a=(B)b;</code>
10	<code> a.f();</code>
11	<code> }}</code>

Test 2 java A

00	<code>class B{</code>
01	<code> B(){ System.out.print("C");}</code>
02	<code> void f() { System.out.print("D");}</code>
03	<code>}</code>
04	<code>public class A extends B {</code>
05	<code> A(){ System.out.print("A");}</code>
06	<code> void f() { System.out.print("B");}</code>
07	<code> public static void main(String k[]) {</code>
08	<code> B b=new B();</code>
09	<code> A a=(A)b;</code>
10	<code> a.f();</code>
11	<code> }}</code>

Test 3

A00	<code>package esame; // NOTA :QUESTA CLASSE E' NEL FILE A.java</code>
A01	<code>public class A {</code>
A02	<code> int x=3;</code>
A03	<code> public static void main(String string[]) {</code>
A04	<code> (new abcd.B()).f();</code>
A05	<code> } }</code>
B01	<code>package abcd; // NOTA :QUESTA CLASSE E' NEL FILE B.java</code>
B02	<code>public class B extends esame.A{</code>
B03	<code> public void f(){</code>
B04	<code> System.out.println(++x);</code>
B05	<code> } }</code>

Test 4

01	<code>#include <stdlib.h></code>
02	<code>#include <iostream.h></code>
03	<code>Using namespace std;</code>
04	<code>void cambia(int m, int n[7]) {</code>
05	<code> (*(n+m))--; m++; n--;</code>
06	<code>}</code>
07	<code>int main(){</code>
08	<code> int vet[] = {1,2,3,4,5};</code>
09	<code> cambia(vet[4],vet-1);</code>
10	<code> cambia(0,&vet[4]);</code>
11	<code> int i=0;</code>
12	<code> for (i=0;i<5;i++) cout << vet[i];</code>
13	<code> return 0;</code>
14	<code>}</code>

A

Test 5 java Main

01	<code>class A{</code>
02	<code> public static int x;</code>
03	<code> A(int s) {x=s;}</code>
04	<code> void f() {System.out.print(x);}</code>
05	<code> public static void main(String a[]){</code>
06	<code> A b=new A(3);</code>
07	<code> A c=new A(4);</code>
08	<code> b.f();</code>
09	<code> c.f();</code>
10	<code> }</code>
11	<code>}</code>

Test 6 java B

00	<code>class B {</code>
01	<code> static int s=0;</code>
02	<code> B(int i){s=i;}</code>
05	<code> public static void main(String[] args) {</code>
06	<code> B b1=new B(3);</code>
07	<code> B b2=new B(3);</code>
08	<code> B b3=new B(1);</code>
09	<code> if (b1.equals(b2)) System.out.print("A"); else</code> <code> System.out.print("B");</code>
10	<code> if (b1.s==b3.s) System.out.print("D"); else</code> <code> System.out.print("C");</code>
11	<code> } }</code>

Test 7 java Prova

01	<code>class A {</code>
02	<code> A(int x) {System.out.print("X");}</code>
03	<code> A() {System.out.print("Z");}</code>
04	<code> public void finalize() {System.out.print("Y");}</code>
05	<code>}</code>
06	<code>class B extends A {</code>
07	<code> B(int x) {System.out.print("A");}</code>
08	<code> B() {System.out.print("C");}</code>
09	<code> public void finalize() {System.out.print("B");}</code>
10	<code>}</code>
11	<code>public class Prova {</code>
12	<code> public static void main(String args[]) {</code>
13	<code> A a=new B(3);</code>
14	<code> a = null;</code>
15	<code> System.gc();</code>
16	<code> System.runFinalization();</code>
17	<code> }</code>
18	<code>}</code>

A

Test 8

01	package uno;
02	public class A {
03	void f(int k) {
04	System.out.print(k*3);
05	}
06	public static void main (String args[]){
07	Object z = new B();
08	if (z instanceof uno.A) ((A) z).f(3);
09	if (z instanceof uno.B) ((B) z).f(3);
10	}
11	}
12	class B extends A{
13	void f(int k) {
14	System.out.print(k*2);
15	}
16	}

Test 9 – scrivere nel campo per l’output del test la sequenza risultante indicando T per le affermazioni vere e F per quelle false

9.1	Non è possibile istanziare una Collection
9.2	In una classe ci può essere un solo metodo main
9.3	In un programma ci può essere un solo metodo public static void main(String[] args)
9.4	In Java esiste ereditarietà multipla
9.5	Java usa solo la heap perché tutta la memoria è allocata dinamicamente con le new()
9.6	Una Interface non può ereditare da una classe
9.7	Se di un metodo f faccio overriding non è detto che ci siano più “signatures” (firme) di metodi legate al nome f
9.8	Se di un metodo f faccio overloading non è detto che ci siano più “signatures” (firme) di metodi legate al nome f