1. What is the output for the below code?

public class A {

int add(int i, int j){

return i+j;

}

}

public class B extends A{

public static void main(String argv[]){

short s = 9;

System.out.println(add(s,6));

}

}

Hint: Static Reference

Ans. The code will not compile as you cannot reference non static methods from static methods.

Main.java:10: error: non-static method add(int,int) cannot be referenced from a static context

1. What is the output for the below code ?

public class A {

int k;

boolean istrue;

static int p;

public void printValue() {

System.out.print(k);

System.out.print(istrue);

System.out.print(p);

}

}

public class Test{

public static void main(String argv[]){

A a = new A();

a.printValue();

}

}

Hint: Default Value for Variables

Ans. The code will produce output as follows : **0false0** (Because print is used instead of println, the code will output on the same line )

Default values for k is 0, istrue is 0 ie. False and p is 0.

1. What is the output for the below code ?
2. What is the output for the below code ?

package com;

class Animal {

public void printName(){

System.out.println("Animal");

}

}

package exam;

import com.Animal;

public class Cat extends Animal {

public void printName(){

System.out.println("Cat");

}

}

package exam;

import com.Animal;

public class Test {

public static void main(String[] args){

Animal a = new Cat();

a.printName();

}

}

Hint: Modifiers

Ans. The code will not compile as the Class Animal which is being inherited by Class Cat is not declared as public. Therefore the class Animal will not be visible as it is declared with the default access specifier. Hence, it will not be visible outside of the package.

Error : Animal cannot be resolved to a type

1. What is the output for the below code ?

public class A {

int i = 10;

public void printValue() {

System.out.println("Value-A");

};

}

public class B extends A{

int i = 12;

public void printValue() {

System.out.print("Value-B");

}

}

public class Test{

public static void main(String argv[]){

A a = new B();

a.printValue();

System.out.println(a.i);

}

}

Hint:Polymorphism

Ans. Here, in this code we are using the run time polymorphism ie. Method overriding. The method printValue is being overridden in the child class B which inherits the same method from the parent class A.

Therefore, the output of the code will be Value-B10. Here 10 is being printed because we are declaring the instance of Class B using the type definition of parent class A. Hence the value of variable I will be 10.

Output : Value-B10

1. What is the output for the below code ?

What is the output for the below code ?

public class A {

static{System.out.println("static");}

{ System.out.println("block");}

public A(){

System.out.println("A");

}

public static void main(String[] args){

A a = new A();

}

Hint: Execution Sequence

Ans. Here the code will output as follows :

static

block

A

Because order of execution is 1. Static Block 2. Instance Block 3. Constructor

1. What is the output for the below code ?

public class Test {

public static void main(String[] args){

byte b = 6;

b+=8;

System.out.println(b);

b = b+7;

System.out.println(b);

}

}

Hint: Casting

Ans. Here, if we run the code as it is given the code will output error as we are directly adding int to the byte variable b. We will have to add type cast as b = (byte) b + 7 for it to work.

Error : Type mismatch: cannot convert from int to byte

1. What is the output for the below code ?

public static void main(String[] args){

String value = "abc";

changeValue(value);

System.out.println(value);

}

public static void changeValue(String a){

a = "xyz";

}

}

Hint: Pass by reference and Value

Ans. Here the code will output abc.

Output : **abc**

1. What is the output for the below code ?

public class A {

public void printValue(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printNameB(){

System.out.println("Name-B");

}

}

public class C extends A{

public void printNameC(){

System.out.println("Name-C");

}

}

public class Test{

public static void main (String[] args) {

B b = new B();

C c = new C();

newPrint(b);

newPrint(c);

}

public static void newPrint(A a){

a.printValue();

}

Hint: Inheritance and Modifiers

Ans. Here the code will output

Value-A

Value-A

As both the classes B and C inherit the class A.

1. What is the output for the below code ?

public class A {

public void printName(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printName(){

System.out.println("Name-B");

}

}

public class C extends A{

public void printName(){

System.out.println("Name-C");

}

}

public class Test{

public static void main (String[] args) {

B b = new B();

C c = new C();

b = c;

newPrint(b);

}

public static void newPrint(A a){

a.printName();

}

}

Hint: Referencing Objects (Super type /Sub types)

Ans: Compilation error because b and c are different classes which is not related. Therefore can not be assign to each other.

1. What is the output for the below code ?

public class C {

}

public class D extends C{

}

public class A {

public C getOBJ(){

System.out.println("class A - return C");

return new C();

}

}

public class B extends A{

public D getOBJ(){

System.out.println("class B - return D");

return new D();

}

}

public class Test {

public static void main(String... args) {

A a = new B();

a.getOBJ();

}

}

Hint: Compatible Return Type

Output : **class B - return D**

1. What is the output for the below code ?

public class A {

private void printName(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printName(){

System.out.println("Name-B");

}

}

public class Test{

public static void main (String[] args) {

B b = new B();

b.printName();

}

}

Hint: overriding – Private Methods

Ans. Here code will output **Name-B**

1. What is the output for the below code ?

What is the output for the below code ?

public class A {

public A(){

System.out.println("A");

}

public A(int i){

this();

System.out.println(i);

}

}

public class B extends A{

public B (){

System.out.println("B");

}

public B (int i){

this();

System.out.println(i+3);

}

}

public class Test{

public static void main (String[] args){

new B(5);

}

}

Hint: Constructor OverLoading

Here the code will output :

A

B

8

1. What is the output for the below code ?

public class D {

int i;

int j;

public D(int i,int j){

this.i=i;

this.j=j;

}

public void printName() {

System.out.println("Name-D");

}

}

public class Test{

public static void main (String[] args){

D d = new D();

d.printName();

}

}

Hint: Constructors

Ans. Here the code will output Error as the default constructor for the class D is not defined. Because we have already defined a parameterized constructor the compiler will not add a normal constructor ie. Default constructor into the code as that is done only when no Constructor is defined for the class. Hence, the output will be an error saying The constructor D() is undefined