**Programs:**

1.

public class Main {

int a=10, b=30;

public void add(){

System.out.println(a+b);

}

public static void main(String[] args) {

add();

}

}

Output: CE: As we are trying to use the add method which is non-static in nature inside the main method which is static in nature

Solution: Either make the add method as static or create an object of the class contains add method and access the add method using the object reference

2.

public class Main {

static int a=10, b=30;

public static void add(){

System.out.println(a+b);

}

public static void main(String[] args) {

add();

}

}

Output: 40

3.

public class Main {

int a=10, b=30;

public void add(){

System.out.println(a+b);

}

public static void main(String[] args) {

Main m = new Main();

m.add();

}

}

Output: 40

4.

class Add1{

int a=10, b=40;

public void add(){

System.out.println(a+b);//50

}

}

public class Main {

public static void main(String[] args) {

Add1 m = new Add1();

m.add();

}

}

Output: 50

5.

class Add1{

int a=10, b=40;

}

public class Main {

public static void main(String[] args) {

Add1 m = new Add1();

m.add();

}

}

Output: CE as add method doesn’t exist inside the Add class

6.

package class\_methods;

import java.util.Scanner;

class Add260401{

public void add(int a, int b){

System.*out*.println(a+b);

}

}

public class AddMain260401 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

System.*out*.println("Enter the first number: ");

int a = scanner.nextInt();

System.*out*.println("Enter the second number: ");

int b = scanner.nextInt();

Add260401 a1 = new Add260401();

a1.add(a, b);

}

}

Output: If a from main method is equals to 10 and b=20 then we’ll get 30 as output. Here the variable a, b from main method from AddMain260401 are not equals to the variable a, b there for add method from Add260401 class as a, b variable from main method are local to the main method only and they can’t be used outside of the main method at all. Similarly for a, b for add method from Add260401 class.

7.

package class\_methods;

class FindEven{

public boolean isEven(int a){

int res = a%2; *//We are collecting the remainder*

boolean isEven = res==0; *//Comparing the remainder value with 0*

return isEven;

}

}

public class Main260402 {

public static void main(String[] args) {

FindEven fe = new FindEven();

fe.isEven(34);

}

}

Output: We will not get any output printed in the console. Reason: We are invoking the isEven method from Main class main method which is returning the boolean value and we are not storing them any where that’s why we can’t use the same value/returned value further

8.

package class\_methods;

class FindEven1{

public boolean isEven(int a){

int res = a%2; *//We are collecting the remainder*

boolean isEven = res==0; *//Comparing the remainder value with 0*

return isEven;

}

}

public class Main260403 {

public static void main(String[] args) {

FindEven fe = new FindEven();

System.*out*.println(fe.isEven(34));

}

}

Output: true as after invoking the isEven method using fe reference, we are trying to print the value on the fly.

9.

package class\_methods;

class FindEven2{

public boolean isEven(int a){

int res = a%2; *//We are collecting the remainder*

boolean isEven = res==0; *//Comparing the remainder value with 0*

return isEven;

}

}

public class Main260403 {

public static void main(String[] args) {

FindEven fe = new FindEven();

boolean isEven = fe.isEven(33);

if (isEven)

System.*out*.println("Even");

else

System.*out*.println("Odd");

}

}

Output: Odd. In this case we are storing in isEven variable inside the main method which is returned from isEven method from FindEven2 class

10.

package class\_methods;

class FindEven3{

public boolean isEven(int a){

return a%2==0;//Here we calculated and returned the value

}

}

public class Main260403 {

public static void main(String[] args) {

FindEven fe = new FindEven();

boolean isEven = fe.isEven(33);

if (isEven)

System.*out*.println("Even");

else

System.*out*.println("Odd");

}

}

Output: Odd.

Note:

1. Why the main method is void?

Ans: main method never returns any value