Short cuts:

1. main --> Control + Space bar + Enter

2. Syso-- Control + Space bar

3. to get options in eclipse press Control + 1

4. To format the code in eclipse press control + Shift + F

5. To comment code in eclipse we use control + forward slash

Rules to design and develop methods: THe main purpose of methods is to reuse the code. Methods makes

our program more organized and structured

Rule 1: Always execution of the program begins with opening bracket of main method

Rule 2: When user defined method calling statement executes/runs then transfer the control to the

matching method

Rule 3: When closing bracket of user defined method runs, transfer the control back to calling

statement

Rule 4: When closing bracket of main method runs complete program stops

Note:

Rule 1 and rule 4 will execute only once in the program

Rule 2 and rule 3 can execute any no. of times

Note:

1. when in argument we use thre dots ... it will let the variable stire any number of values

2. var datatype cannot used for method arguments

Example 1:

public class A {

public static void main(String[] args) {//Rule 1 START

A a1 = new A();//No

a1.test();//Rule 2

}//Rule 4

private void test() {//Control comes here

System.out.println(1000);//NO

}//Rule 4

}

//Output 1000

Example 2:

public class A {

public static void main(String[] args) {//Rule 1 START

A a1 = new A();//No

a1.test();//Rule 2

System.out.println(5);//NO

a1.test();//Rule 2

}//Rule 4 STOP

public void test() {//Control Comes Here

System.out.println(1000);

}//Rule 3 Rule 3

}

//Output 1000 5 1000

Example 3:

public class A {

public static void main(String[] args) {//Rule 1 STARTS here

A a1 = new A(); //No Rule

a1.test1();//Rule 2

System.out.println(5);//No

}//Rule 4 STOP

public void test1() {//Control Comes here

System.out.println(1000);

A a2 = new A();//No

a2.test2();//Rule 2

}//Rule 3

public void test2(){

System.out.println("From test 2");

}//Rule 3

}

//Output: 1000 From test 2 5

Example 4:

public class A {

public static void main(String[] args) {//Rule 1 START here

A a1 = new A(); //No

a1.test1();//Rule 2

System.out.println(5);//No

A a2 = new A();//No

a2.test2();//Rule 2

}//Rule 4 STOP

public void test1() {//Control COmes here

System.out.println(1000);//NO

}//Rule 3

public void test2() {//Control Comes here

System.out.println(500);//No

}//Rule 3

}

Output:

1000

5

500

Example 5:

Note: Arguments created in methods are treated as local variables:

public class A {

public static void main(String[] args) {//Rule 1 START here

A a1 = new A();

a1.test1(100);

}

public void test1(int i) {

System.out.println(i);

}

}

Output:

100

Example 6:

public class A {

public static void main(String[] args) {//Rule 1 START here

A a1 = new A();

a1.test1(100,200,300,400);

}

public void test1(int... i) {

System.out.println(i[0]);

System.out.println(i[1]);

System.out.println(i[2]);

System.out.println(i[3]);

}

}

Output:

100

200

300

400

Example 7:

public class A {

public static void main(String[] args) {//Rule 1 START here

A a1 = new A();

a1.test1(100, "Pankaj Sir Academy", true);

}

public void test1(var... i) {

System.out.println(i[0]);

System.out.println(i[1]);

System.out.println(i[2]);

System.out.println(i[3]);

}

}

Output:

Error