CSA0994 – PROGRAMMING IN JAVA FOR DUMMIES

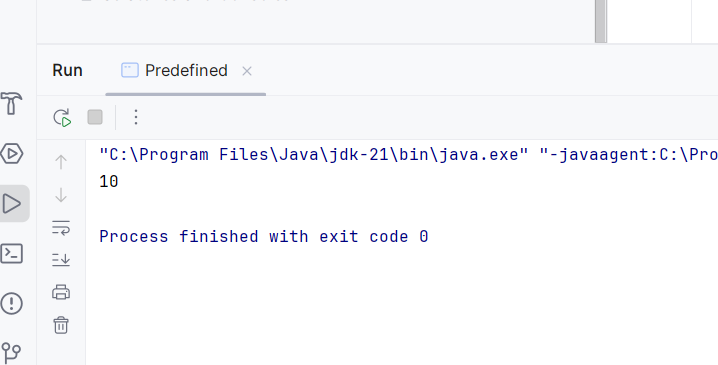
Day -1 Programs

Program 1: Predefined Functions

Code:

public class Predefined  
{  
 void add( int a, int b)  
 {  
 int c;  
 a=5;  
 b=5;  
 c=a+b;  
 System.*out*.print(c);  
 }  
 public static void main1(String args[])  
 {  
 Predefined p= new Predefined();  
 p.add( 8,9);  
 }  
}

Output:

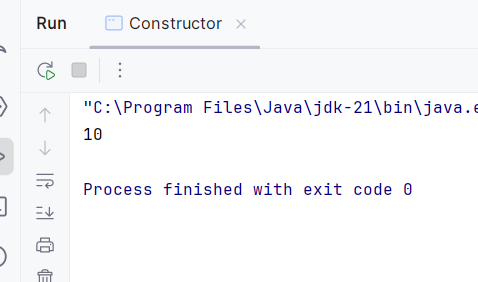


Program 2: Constructor

Code:

public class Constructor {  
 int a,b;  
 Constructor() {  
  
 a=5;  
 b=5;  
  
 }  
 void add()  
 {  
 int c;  
 c=a+b;  
 System.*out*.println(c);  
  
 }  
 public static void main(String[] args)  
 {  
 Constructor p=new Constructor();  
 p.add();  
 }  
}

Output:



Program 3 : Static

Code:

public class Static\_method {

static int add()

{

int a=10;

int b=5;

int c= a+b;

return c;

}

public static void main(String[] ar)

{

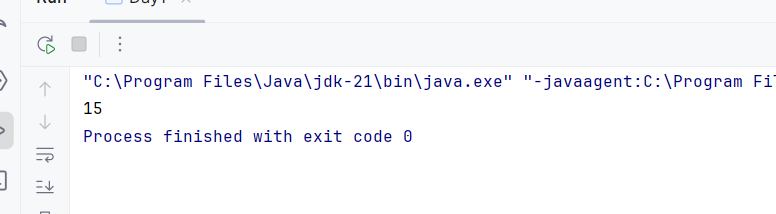
int n = Static\_method.add();

System.out.println("Sum :"+n);

}

}

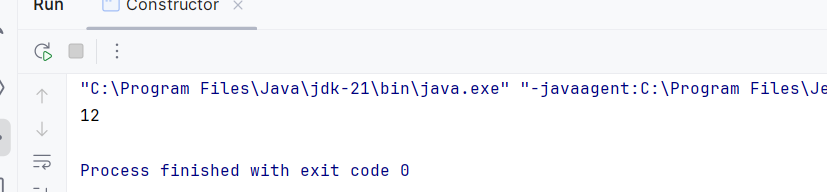
Output:



Program 4: Final

Code:

public class Constructor {  
 int a,b;  
 Constructor() {  
  
 a=5;  
 b=6;  
  
 }  
 void add()  
 {  
 int c;  
 final int i=10;  
 c=a+b;  
 System.*out*.println(c);  
  
 }  
 public static void main(String[] args)  
 {  
 Constructor p=new Constructor();  
 p.add();  
 }  
}

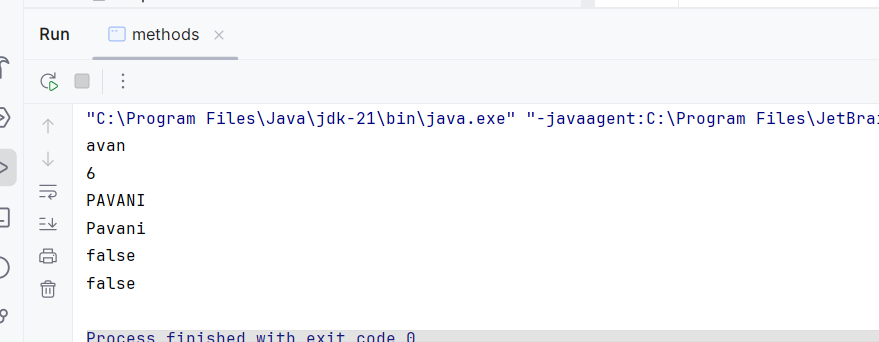
Output: 

Program 5: Methods in String

Code:

import java.util.Locale;  
public class methods  
{  
 public static void main(String args[])  
 {  
 String s="Pavani";  
 String g=s.toUpperCase();  
 String h=s.toLowerCase();  
 String i=s.substring(1,5);  
 System.*out*.println(i);  
 System.*out*.println(s.length());  
 System.*out*.println(g);  
 System.*out*.println(s.strip());  
 System.*out*.println(s.isBlank());  
 System.*out*.println(s.isEmpty());  
  
 }  
}

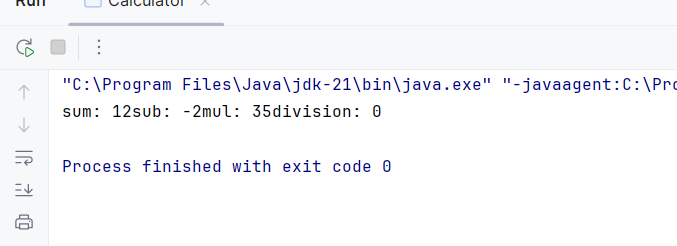
Output:



Program-6: create a class with four methods add sub multi divide call all the four functions from main function within one statement

Code:

import java.util.\*;  
  
public class Calculator {  
 int add()  
 {  
 int a,b,c;  
 a=5;  
 b=7;  
 c=a+b;  
 return c;  
 }  
 int sub()  
 {  
 int a,b,c;  
 a=5;  
 b=7;  
 c=a-b;  
 return c;  
 }  
 int mul() {  
 int a, b, c;  
 a = 5;  
 b = 7;  
 c = a \* b;  
 return c;  
 }  
 int div() {  
 int a, b, c;  
 a = 5;  
 b = 7;  
 c = a/b;  
 return c;  
 }  
 public static void main(String[] ar)  
 {  
 Calculator p = new Calculator();  
 int x= p.add();  
 int y= p.sub();  
 int z= p.mul();  
 int v= p.div();  
 System.*out*.println("sum: "+x+"sub: "+y+"mul: "+z+"division: "+v);  
  
 }  
}

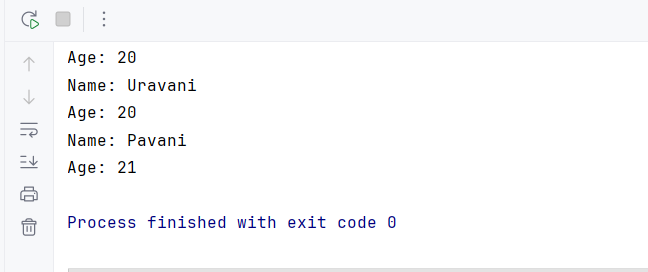
Output: 

Program 7:Overriding

Code:

public class overriding {  
 class Info {  
 int age = 20;  
 String pname = "Uravani";  
  
 {  
 System.*out*.println("Name: " + pname);  
 System.*out*.println("Age: " + age);  
 }  
 }  
 class Derived extends Info {  
 int age = 21;  
 String pname = "Pavani";  
  
 {  
 System.*out*.println("Name: " + pname);  
 System.*out*.println("Age: " + age);  
 }  
 }  
 public static void main(String[] args) {  
 overriding outer = new overriding();  
 Info base = outer.new Info();  
 Derived derived = outer.new Derived();  
 }  
}

Output:

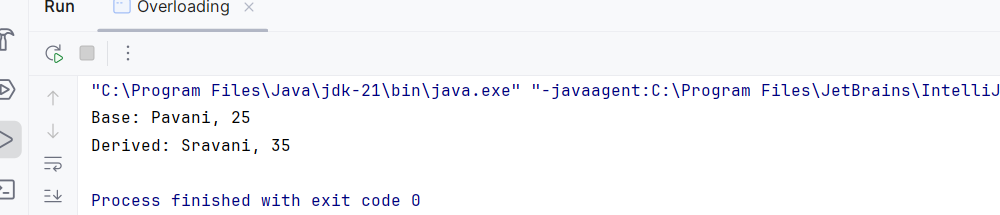


Program 8: Overloading type-1 Decreasing or increasing

Code:

public class Overloading {  
 class Info {  
 int age;  
 String pname;  
  
 void setValues(int age, String pname) {  
 this.age = age;  
 this.pname = pname;  
 System.*out*.println("Base: " + pname + ", " + age);  
 }  
 }  
  
 class Derived extends Info {  
 @Override  
 void setValues(int age, String pname) {  
 this.age = age + 5;  
 this.pname = pname;  
 System.*out*.println("Derived: " + this.pname + ", " + this.age);  
 }  
 }  
  
 public static void main(String[] args) {  
 Overloading outer = new Overloading();  
 Info base = outer.new Info();  
 base.setValues(25, "Pavani");  
 Info derivedAsBase = outer.new Derived();  
 derivedAsBase.setValues(30, "Sravani");  
 }  
}

Output:

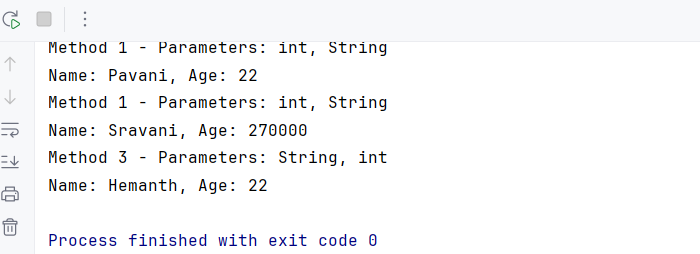


Program 9:Overloading changing Datatype

Code:

public class OverloadingDatatype {  
  
 void setValues(int age, String pname) {  
 System.*out*.println("Method 1 - Parameters: int, String");  
 System.*out*.println("Name: " + pname + ", Age: " + age);  
 }  
  
 void setValues(double salary, String pname) {  
 System.*out*.println("Method 2 - Parameters: double, String");  
 System.*out*.println("Name: " + pname + ", Salary: " + salary);  
 }  
  
 void setValues(String pname, int age) {  
 System.*out*.println("Method 3 - Parameters: String, int");  
 System.*out*.println("Name: " + pname + ", Age: " + age);  
 }  
  
 public static void main(String[] args) {  
 OverloadingDatatype obj = new OverloadingDatatype();  
  
 obj.setValues(22, "Pavani");  
 obj.setValues(270000, "Sravani");  
 obj.setValues("Hemanth", 22);  
 }  
}

Output:



Program 10: Overloading type – 3 interchanging

Code:

public class OverloadingInterchange {  
 void setValues(int age, String pname) {  
 System.*out*.println("Age: " + age + ",Name:" + pname);  
 }  
 void setValues(String pname, int age) {  
   
 System.*out*.println("Name: " + pname + ", Age: " + age);  
 }  
 public static void main(String[] args) {  
 OverloadingInterchange obj = new OverloadingInterchange();  
 obj.setValues(22, "Hemanth");  
 obj.setValues("Pavani", 21);  
 }  
}

}

Output:

