Swapnanil Dutta Roll: 12

Assignment 4

Question 2:

Code:

class Outer {

String name = "Miss Dawson Homenick";

void show() {

System.out.println("This is the outer class");

System.out.println("Name : " + name);

}

public class Inner {

String occupation = "Executive";

void display() {

System.out.println("This is the inner class");

show();

System.out.println("Occupation :" + occupation);

}

}

public static void main(String[] args) {

new Outer().show();

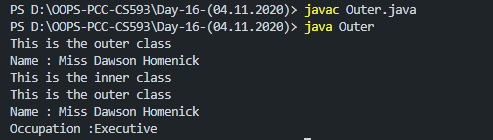
new Outer().new Inner().display();

// new Outer().display(); ! Shows error

}

}

Output:



Question 4:

Code:

class AnonymousClass {

void show() {

System.out.println("Super class method");

}

}

interface A1 {

public AnonymousClass get();

}

class B1 implements A1 {

public AnonymousClass get() {

return new AnonymousClass() {

void show() {

System.out.println("Overridden method");

}

};

}

}

class Ass4Q4 {

public static void main(String[] args) {

new B1().get().show();

}

}

Output:



Question 5:

Code:

class Block1 {

public Block1() {

System.out.println("Block1 Constructor");

}

// Block1 Initializer block

{

System.out.println("Block1 Initializer Block1");

}

{

System.out.println("Block1 Initializer Block2");

}

}

class Block2 extends Block1 {

public Block2() {

System.out.println("Block2 Constructor");

}

{

System.out.println("Block2 Initializer Block1");

}

{

System.out.println("Block2 Initializer Block2");

}

}

public class BlockTest1 {

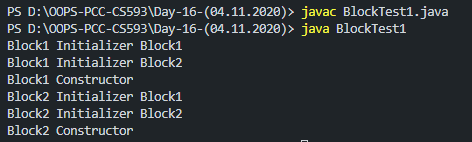
public static void main(String[] args) {

new Block2();

}

}

Output:



Question 6:

Code:

class A {

A() {

System.out.println("Super constructor");

}{

System.out.println("Super Initializer block");

}// Super Initializer block

static {

System.out.println("Super static block");

}

}

// static blocks are called when a class is loaded into jvm

class B extends A {

B() {

System.out.println("Sub constructor");

}

// Sub Initializer block

{

System.out.println("Sub Initializer block");

}

static {

System.out.println("Sub static block");

}

}

public class BlockTest2 {

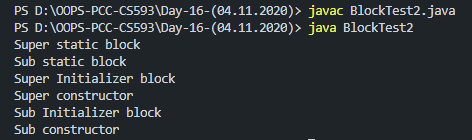
public static void main(String[] args) {

new B();

}

}

Output:



Question 7:

Code:

public class BlockTest3 {

public static void main(String[] args) {

new Block2();

}

}

class Block2 {

static int a;

int b;

Block2() {

System.out.println("Constructor Block");

}

static {// only static fields can be accessed inside static blocks

a = 40;

System.out.println("a=" + a);

}

{

// both static and non-static fields can be accessed inside intializer blocks

a = 30;

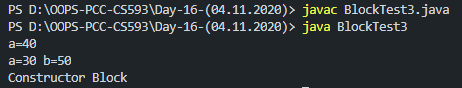
b = 50;

System.out.println("a=" + a + " b=" + b);

}

}

Output:



Question 8:

Code:

class A1 {

A1() {

System.out.println("Super constructor");

}

{

System.out.println("Super Initializer block");

}

static {

System.out.println("Super static block");

}

}

// static blocks are called when a class is loaded into jvm

class B1 extends A1 {

B1() {

System.out.println("Sub constructor");

}

{

System.out.println("Sub Initializer block");

}

static {

System.out.println("Sub static block");

}

}

public class BlockTest4 {

public static void main(String[] args) {

new B1();

}

}

Output:

