Questions

1. What is the output of this program?

class increment {

public static void main(String args[])

{

double var1 = 1 + 5;

double var2 = var1 / 4;

int var3 = 1 + 5;

int var4 = var3 / 4;

System.out.print(var2 + " " + var4);

}

}

1. public class Test {

static void test(float x){

System.out.print("float");

}

static void test(double x){

System.out.print("double");

}

public static void main(String[] args){

test(99.9);

}

}

1. class Main {

public static void main(String args[]){

final int i;

i = 20;

System.out.println(i);

}

}

1. class Base {

public final void show() {

System.out.println("Base::show() called");

}

}

class Derived extends Base {

public void show() {

System.out.println("Derived::show() called");

}

}

public class Main {

public static void main(String[] args) {

Base b = new Derived();;

b.show();

}

}

1. What is the output of this program?

class increment {

public static void main(String args[])

{

int g = 3;

System.out.print(++g \* 8);

}

}

1. What is the output of the following code?

class San

{

public void m1 (int i,float f)

{

System.out.println(" int float method");

}

public void m1(float f,int i);

{

System.out.println("float int method");

}

public static void main(String[]args)

{

San s=new San();

s.m1(20,20);

}

} ``````````

1. What is the output of this program?

class overload {

int x;

int y;

void add(int a) {

x = a + 1;

}

void add(int a, int b){

x = a + 2;

}

}

class Overload\_methods {

public static void main(String args[])

{

overload obj = new overload();

int a = 0;

obj.add(6);

System.out.println(obj.x);

}

}

1. class test {

int a;

int b;

test(int a, int b) {

a = a;

b = b;

}

void meth(test o) {

o.a \*= 2;

o.b /= 2;

}

}

public class Output {

public static void main(String args[])

{

test obj = new test(10 , 20);

obj.meth(obj);

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

}

}

1. What is the output of this program?

class Output {

public static void main(String args[])

{

int a = 1;

int b = 2;

int c;

int d;

c = ++b;

d = a++;

c++;

b++;

++a;

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

}

}

1. What is the output of this program?

class A {

int i;

}

class B extends A {

int j;

void display() {

super.i = j + 1;

System.out.println(j + " " + i);

}

}

class inheritance {

public static void main(String args[])

{

B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}



class Test {

     public static void main(String args[])  {

       System.out.println(10  +  20 + "GeeksQuiz");

       System.out.println("GeeksQuiz" + 10 + 20);

   }

}

1. class Derived extends Base {

   public static void main(String args[]){

      Base a = new Derived();

      System.out.println(a instanceof Derived);

   }

}

True

The instanceof operator works even when the reference is of base class type.

1. class Test

{

    public static void main(String args[])

    {

        String s1 = "geeksquiz";

        String s2 = "geeksquiz";

        System.out.println("s1 == s2 is:" + s1 == s2);

    }

}

1. Predict the output of the following program

class Test

{

    boolean[] array = new boolean[3];

    int count = 0;

    void set(boolean[] arr, int x)

    {

        arr[x] = true;

        count++;

    }

    void func()

    {

        if(array[0] && array[++count - 2] | array [count - 1])

            count++;

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

    }

    public static void main(String[] args)

    {

        Test object = new Test();

        object.set(object.array, 0);

        object.set(object.array, 1);

        object.func();

    }

}

1. class Test {

int i;

}

public class Main {

public static void main(String args[]) {

Test t ;

System.*out*.println(t.i);

}

}

1. class Test {

int i;

}

public class Main {

public static void main(String args[]) {

Test t=null ;

System.*out*.println(t.i);

}

}

class Test {

int i;

}

public class Main {

public static void main(String args[]) {

try{

Test t = null ;

System.*out*.println(t.i);

}catch(Exception e){

System.*out*.println(e.getMessage());

}

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

}

}

1. Predict the output of following Java program

|  |
| --- |
|  |

class Test {

int i;

}

class Main {

public static void main(String args[]) {

Test t = new Test();

System.out.println(t.i);

}

}

1. class demo

{

    int a, b;

    demo()

    {

        a = 10;

        b = 20;

    }

    public void print()

    {

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

    }

}

class Test

{

    public static void main(String[] args)

    {

        demo obj1 = new demo();

        demo obj2 = obj1;

        obj1.a += 1;

        obj1.b += 1;

        System.out.println ("values of obj1 : ");

        obj1.print();

        System.out.println ("values of obj2 : ");

        obj2.print();

    }

}

1. Predict the output of the following program.

|  |
| --- |
| class First  {        void display()      {          System.out.println("Inside First");      }  }    class Second extends First  {        void display()      {          System.out.println("Inside Second");      }  }      class Test  {        public static void main(String[] args)      {          First obj1 =  new First();          Second obj2 =  new Second();            First ref;          ref = obj1;          ref.display();            ref = obj2;          ref.display();      }  } |

1. Predict the output of following Java program

|  |
| --- |
| // Note static keyword after import.  import static java.lang.System.\*;    class StaticImportDemo  {     public static void main(String args[])     {          out.println("GeeksforGeeks");     }  } |

**22.** Predict the output of following Java program

|  |
| --- |
| class Main {     public static void main(String args[]) {        try {           throw 10;        }        catch(int e) {           System.out.println("Got the  Exception " + e);        }    }  } |

1. class Test extends Exception { }

class Main {

   public static void main(String args[]) {

      try {

         throw new Test();

      }

      catch(Test t) {

         System.out.println("Got the Test Exception");

      }

      finally {

         System.out.println("Inside finally block ");

      }

  }

}

1. Output of following Java program?

|  |
| --- |
| class Main {     public static void main(String args[]) {        int x = 0;        int y = 10;        int z = y/x;    }  } |

1. class Base extends Exception {}

class Derived extends Base  {}

public class Main {

  public static void main(String args[]) {

   // some other stuff

   try {

       // Some monitored code

       throw new Derived();

    }

    catch(Base b)     {

       System.out.println("Caught base class exception");

    }

    catch(Derived d)  {

       System.out.println("Caught derived class exception");

    }

  }

}

1. class Test

{

    public static void main (String[] args)

    {

        try

        {

            int a = 0;

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

            int b = 20 / a;

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

        }

        catch(ArithmeticException e)

        {

            System.out.println ("Divide by zero error");

        }

        finally

        {

            System.out.println ("inside the finally block");

        }

    }

}

1. class Test

{

    public static void main(String[] args)

    {

        try

        {

            int a[]= {1, 2, 3, 4};

            for (int i = 1; i <= 4; i++)

            {

                System.out.println ("a[" + i + "]=" + a[i] + "\n");

            }

        }

        catch (Exception e)

        {

            System.out.println ("error = " + e);

        }

        catch (ArrayIndexOutOfBoundsException e)

        {

            System.out.println ("ArrayIndexOutOfBoundsException");

        }

    }

}

Predict the output of the following program.

|  |
| --- |
| class Test  {      String str = "a";        void A()      {          try          {              str +="b";              B();          }          catch (Exception e)          {              str += "c";          }      }        void B() throws Exception      {          try          {              str += "d";              C();          }          catch(Exception e)          {              throw new Exception();          }          finally          {              str += "e";          }            str += "f";        }        void C() throws Exception      {          throw new Exception();      }        void display()      {          System.out.println(str);      }        public static void main(String[] args)      {          Test object = new Test();          object.A();          object.display();      }    } |

1. Predict the output of the following program.

|  |
| --- |
| class Test  {   int count = 0;        void A() throws Exception      {          try          {              count++;                try              {                  count++;                    try                  {                      count++;                      throw new Exception();                    }                    catch(Exception ex)                  {                      count++;                      throw new Exception();                  }              }                catch(Exception ex)              {                  count++;              }          }            catch(Exception ex)          {              count++;          }        }        void display()      {          System.out.println(count);      }        public static void main(String[] args) throws Exception      {          Test obj = new Test();          obj.A();          obj.display();      }  } |