1. The following Java applications contain errors. Point out the statement(s) that contain errors. Explain what each of the errors is, and how it can be fixed.

|  |  |
| --- | --- |
| public class **OOPExercises** {  public static void main(String[] args) {  A objA = new A();  System.out.println("in main(): ");  System.out.println("objA.a = "+objA.a);  objA.a = 222;  }  } | **Point out the error(s) and how they can be fixed.** |
| public class **A** {  private int a = 100;  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A |

|  |  |
| --- | --- |
| public class **OOPExercises** {  public static void main(String[] args) {  System.out.println("in main(): ");  System.out.println("objA.a = "+getA() );  setA(123);  }  } | **Point out the error(s) and how they can be fixed.** |
| public class **A** {  private int a = 100;  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A |

|  |  |
| --- | --- |
| public class **OOPExercises** {  public static void main(String[] args) {  A objA = new A( );  double result;  result = objA.getA( );  System.out.println("objA.a = "+ result);  }  } | **Point out the error(s) and how they can be fixed.** |
| public class **A** {  private int a = 100;  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A |

|  |  |
| --- | --- |
| public class **B extends A** {  private int a = 222;  public static void main(String[] args) {  System.out.println("in main(): ");  System.out.println("a = "+a );  a = 123;  }  } | **Point out the error(s) and how they can be fixed.** |
| public class **A** {  private int a = 100;  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A |

1. Show the output of the following applications.

|  |  |
| --- | --- |
| public class **OOPExercises** {  public static void main(String[] args) {  A objA = new A();  B objB = new B();  System.out.println("in main(): ");  System.out.println("objA.a = "+objA.getA());  System.out.println("objB.b = "+objB.getB());  objA.setA (222);  objB.setB (333.33);  System.out.println("objA.a = "+objA.getA());  System.out.println("objB.b = "+objB.getB());  }  } | **Output:** |
| public class **A** {  int a = 100;  public A() {  System.out.println("in the constructor of class A: ");  System.out.println("a = "+a);  a = 333;  System.out.println("a = "+a);  }  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A | |
| public class **B** {  double b = 123.45;  public B() {  System.out.println("-----in the constructor of class B: ");  System.out.println("b = "+b);  b = 3.14159;  System.out.println("b = "+b);  }  public void setB( double value) {  b = value;  }  public double getB() {  return b;  }  } //class B | |



|  |  |
| --- | --- |
| public class **OOPExercises** {  public static void main(String[] args) {  //A objA = new A();  B objB = new B();  System.out.println("in main(): ");  //System.out.println("objA.a = "+objA.getA());  System.out.println("objB.b = "+objB.getB());  //objA.setA (222);  objB.setB (333.33);  //System.out.println("objA.a = "+objA.getA());  System.out.println("objB.b = "+objB.getB());  }  } | **Output:** |
| public class **A** {  int a = 100;  public A() {  System.out.println("in the constructor of class A: ");  System.out.println("a = "+a);  a = 333;  System.out.println("a = "+a);  }  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A | |
| public class **B extends A** {  double b = 123.45;  public B() {  System.out.println("-----in the constructor of class B: ");  System.out.println("b = "+b);  b = 3.14159;  System.out.println("b = "+b);  }  public void setB( double value) {  b = value;  }  public double getB() {  return b;  }  } //class B | |



|  |  |
| --- | --- |
| public class **OOPExercises** {  static int a = 555;    public static void main(String[] args) {  A objA = new A();  B objB = new B();  System.out.println("in main(): ");  System.out.println("a = "+a);  a = 444;  System.out.println("objB.a = "+objB.getA());  objA.setA (77777);  objB.rollBackA();  System.out.println("After roll back -----");  System.out.println("a = "+a);  System.out.println("objA.a = "+objA.getA());  System.out.println("objB.a = "+objB.getA());  }  } | **Output:** |
| public class **A** {  int a = 100;  public A() {  //System.out.println("in the constructor of class A: ");  //System.out.println("a = "+a);  a = 333;  //System.out.println("a = "+a);  }  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class A | |
| public class **B** extends A {  private int a = 123;  public B() {  a = 2222;  }  public void rollBackA () {  a = super.getA();  }  public void setA( int value) {  a = value;  }  public int getA() {  return a;  }  } //class B | |



|  |  |
| --- | --- |
| public class **OOPExercises** {  static int a = 555;    public static void main(String[] args) {  A objA = new A();  B objB1 = new B();  A objB2 = new B();  C objC1 = new C();  B objC2 = new C();  A objC3 = new C();  objA.display();  objB1.display();  objB2.display();  objC1.display();  objC2.display();  objC3.display(); }  } | **Output:** |
| public class **A** {  int a = 100;  public void display() {  System.out.printf("a in A = %d\n", a);  }  } //class A | |
| public class **B** extends A {  private int a = 123;  public void display() {  System.out.printf("a in B = %d\n", a);  }  } //class B | |
| public class **C** extends B {  private int a = 543;  public void display() {  System.out.printf("a in C = %d\n", a);  }  } //class C | |

1. **What will be the output of the following program?**

class X

{

    void method(int a)

    {

        System.out.println("ONE");

    }

    void method(double d)

    {

        System.out.println("TWO");

    }

}

class Y extends X

{

    @Override

    void method(double d)

    {

        System.out.println("THREE");

    }

}

public class MainClass

{

    public static void main(String[] args)

    {

        new Y().method(100);

    }

}

4. List out the invalid and valid cases for method overloading with reasons.

4.1 int mymethod(int a, int b, float c)

int mymethod(int var1, int var2, float var3)

4.2 int mymethod(int a, int b)

int mymethod(float var1, float var2)

4.3 int mymethod(int a, int b)

int mymethod(int num)

4.4 float mymethod(int a, float b)

float mymethod(float var1, int var2)

4.5 int mymethod(int a, int b)

float mymethod(int var1, int var2)

5. **What will be the output of the following program?**

class Demo

{

public int myMethod(int num1, int num2)

{

System.out.println("First myMethod of class Demo");

return num1+num2;

}

public int myMethod(int var1, int var2)

{

System.out.println("Second myMethod of class Demo");

return var1-var2;

}

}

class Sample4

{

public static void main(String args[])

{

Demo obj1= new Demo();

obj1.myMethod(10,10);

obj1.myMethod(20,12);

}

}

6. **What will be the output of the following program?**

class Demo2

{

public double myMethod(int num1, int num2)

{

System.out.println("First myMethod of class Demo");

return num1+num2;

}

public int myMethod(int var1, int var2)

{

System.out.println("Second myMethod of class Demo");

return var1-var2;

}

}

class Sample5

{

public static void main(String args[])

{

Demo2 obj2= new Demo2();

obj2.myMethod(10,10);

obj2.myMethod(20,12);

}

}