

LARATECHNOLOGIES

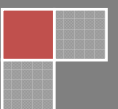
2012

LARA TECHNOLOGIES CORE JAVA PART-2



02-11-2012

102/12, 2nd Main, 5th Cross, Venkateswara College Road, Ramaiah Garden, Chikka
Aduodi, Thavarakere, Bangalore – 560029.
Phone No: 080 – 4131 0124



1. **Program**

```
enum Days
{
    MON, TUE, WED, THRS, FRI,
    SAT, SUN;
}
```

2. **Program**

```
class Manager
{
    public static void main(String[] args)
    {
        Days d1 = Days.MON;
        System.out.println(d1);
        Days d2 = Days.FRI;
        System.out.println(d2);
        Days d3 = Days.SUN;
        System.out.println(d3);
    }
}
```

3. **Program**

```
enum Month
{
    JAN, FEB, MAR, APR, MAY, JUN,
    JUL, AUG, SEP, OCT, NOV, DEC;
}

class Manager1
{
    public static void main(String[] args)
```

```
{
    Month m1 = Month.JAN;
    System.out.println(m1);
    m1 = Month.JUL;
    System.out.println(m1);
    m1 = Month.MAR;
    System.out.println(m1);
}
```

4. **Program**

```
class Manager2
{
    enum A
    {
        CON1, CON2, CON3;
    }

    public static void main(String[] args)
    {
        A a1 = A.CON2;
        System.out.println(a1);
        A a2 = A.CON3;
        System.out.println(a2);
    }
}
```

5. **Program**

```
class A
{
    enum B
    {
```

```

        C1, C2, C3;
    }
    static void test()
    {
        B b1 = B.C2;
        System.out.println(b1);
        B b2 = B.C3;
        System.out.println(b2);
    }
}
class Manager3
{
    public static void main(String[] args)
    {
        A.B b1 = A.B.C2;
        System.out.println(b1);
        A.B b2 = A.B.C3;
        System.out.println(b2);
        A.test();
    }
}

```

6. Program

```

class Manager4
{
    enum E
    {
        a, b, c, d;
    }
    public static void main(String[] args)
    {

```

```

        E e1 = E.b;
        System.out.println(e1);
        E e2 = E.d;
        System.out.println(e2);
        System.out.println(e1.ordinal());
        System.out.println(e2.ordinal());
    } }

```

7. Program

```

class Manager5
{
    enum Test
    {
        t1, t2, t3, t4, t5;
    }
    public static void main(String[] args)
    {
        Test x[] = Test.values();
        for(int i = 0; i < x.length;
            i++)
        {
            System.out.println(x[i]);
        }
    }
}

```

8. Program

```

class Manager6
{
    enum C
    {

```

```

        X, Y, Z;
    }
    public static void main(String[] args)
    {
        C c1 = C.valueOf("X");
        System.out.println(c1);
        C c2 = C.valueOf("Z");
        System.out.println(c2);
        C c3 = C.valueOf("A");
        System.out.println(c3);
    }
}

```

9. Program

```

class Manager7
{
    enum X
    {
        A, B, C, D;
    }
    public static void main(String[] args)
    {
        X x1 = X.C;
        switch(x1)
        {
            case A:
            {
                System.out.println("in A");
                break;
            }
        }
    }
}

```

```

        case B:
        {
            System.out.println("in B");
            break;
        }
        case C:
        {
            System.out.println("in C");
            break;
        }
        case D:
        {
            System.out.println("in D");
            break;
        }
        }
        System.out.println("done");
    }
}

```

10. Program

```

class Manager8
{
    enum E
    {
        CON(90), TEST(30),
        JAVA(90);
        E(int i)
        {
        }
    }
}

```

```

        public static void main(String[] args)
        {
            E e1 = E.TEST;
            System.out.println(e1);
        }
    }

```

11. Program

```

class Manager9
{
    enum A
    {
        CON, TEST(90),
        HELLO("ABC");
        A()
        {
            System.out.println("A()");
        }
        A(int i)
        {
            System.out.println("A(int)");
        }
        A(String s1)
        {
            System.out.println("A(String)");
        }
    }
    public static void main(String[] args)
    {
        A a1 = A.HELLO;
        System.out.println(a1);
    }
}

```

```

    } }

```

12. Program

```

class Manager10
{
    enum Month
    {
        JAN(31), FEB(28),
        MAR(31);
        int days;
        Month(int days)
        {
            this.days = days;
        }
        int getDays()
        {
            return days;
        }
    }
    public static void main(String[] args)
    {
        Month m1 = Month.FEB;
        System.out.println(m1);
        System.out.println(m1.getDays());
        System.out.println(m1.days);
        System.out.println("-----");
        Month m2 = Month.JAN;
        System.out.println(m2);
        System.out.println(m2.getDays());
        System.out.println(m2.days);
        System.out.println("-----");
    }
}

```

```
    } }
```

13. Program

```
class Manager11
{
    enum A
    {
        CON1, CON2,
        CON3
    {
        void test()
        {
            System.out.println("CSCB-test");
        }
    }, CON4, CON5;

    void test()
    {
        System.out.println("test");
    }

}

public static void main(String[] args)
{
    A a1 = A.CON5;
    A a2 = A.CON1;
    A a3 = A.CON3;
    A a4 = A.CON4;
    System.out.println(a1);
    System.out.println(a2);
    System.out.println(a3);
```

```
    System.out.println(a4);
    a1.test();
    a2.test();
    a3.test();
    a4.test();
```

```
}
```

```
}
```

14. Program

```
package pack1;
public class A
{
    public static int i = 20;
    public static void test()
    {
        System.out.println("done");
    }
}
```

15. Program

```
package pack1;
public interface B
{
    int j = 20;
    String k = "abc";
}
```

16. Program

```
package pack1;

public enum C
{
    CON1, CON2, CON3;
}
```

17. Program

```
package pack2;
class Manager
{
    public static void main(String[] args)
    {
        System.out.println(pack1.A.i);
        pack1.A.test();
        System.out.println(pack1.B.j);
        System.out.println(pack1.B.k);
        System.out.println(pack1.C.CON1);
        System.out.println(pack1.C.CON2);
        System.out.println(pack1.C.CON3);
    }
}
```

18. Program

```
package pack2;
import static pack1.A.i;
import static pack1.A.test;
import static pack1.B.j;
import static pack1.B.k;
```

```
import static pack1.C.CON1;
import static pack1.C.CON2;
import static pack1.C.CON3;
class Manager1
{
    public static void main(String[] args)
    {
        System.out.println(i);
        test();
        System.out.println(j);
        System.out.println(k);
        System.out.println(CON1);
        System.out.println(CON2);
        System.out.println(CON3);
    }
}
```

19. Program

```
package pack2;
import static pack1.A.i;
import static pack1.A.test;
import static pack1.B.j;
import static pack1.B.k;
import static pack1.C.CON1;
import static pack1.C.CON2;
import static pack1.C.CON3;
import static java.lang.System.out;
class Manager2
{
    public static void main(String[] args)
```

```

        {
            out.println(i);
            test();
            out.println(j);
            out.println(k);
            out.println(CON1);
            out.println(CON2);
            out.println(CON3);
        }
    }

```

20. Program

```

package pack2;
class Manager3
{
    public static void main(String[] args)
    {
        System.out.println(Integer.MAX_V
ALUE);

        System.out.println(Long.MAX_VAL
UE);
    }
}

```

21. Program

```

package pack2;
import static
java.lang.Integer.MAX_VALUE;

```

```

//import static
java.lang.Long.MAX_VALUE;
import static java.lang.Long.*;
class Manager4
{
    public static void main(String[] args)
    {
        System.out.println(MAX_VALUE);
    }
}

```

22. Program

```

package pack2;
class Manager5
{
    public static void main(String[] args)
    {
        System.out.println("1111");
        System.exit(0);
        System.out.println("2222");
    }
}

```

23. Program

```

package pack2;
import static java.lang.System.out;
import static java.lang.System.exit;
class Manager6
{

```



```

    public static void main(String[] args)
    {
        out.println("1111");
        exit(0);
        out.println("2222");
    }
}

```

24. Program

```

package pack2;
import static java.lang.System.*;
class Manager7
{
    public static void main(String[] args)
    {
        out.println("1111");
        exit(0);
        out.println("2222");
    }
}

```

25. Program

```

package pack2;
import static pack1.A.*;
import pack1.A;
class Manager8
{
    public static void main(String[] args)
    {

```

```

        System.out.println(i);
        test();
        A a1 = new A();
    }
}

```

26. Program

```

package pack2;
import static pack1.B.*;
//import pack1.B;
class Manager9
{
    public static void main(String[] args)
    {
        System.out.println(j);
        System.out.println(k);
        System.out.println(B.k);
        System.out.println(B.j);
        B b1 = null;
    }
}

```

27. Program

```

package pack2;
import static pack1.C.*;
import pack1.C;
class Manager10
{
    public static void main(String[] args)

```

```

    {
        System.out.println(CON1);
        System.out.println(CON2);
        System.out.println(CON3);
        System.out.println(C.CON3);
        C c1 = C.CON1;
    }
}

```

28. Program

```

package pack1;
class Job
{
    String title;
    double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}

class Employee
{
    private int index;
    Job x[];
    Employee(Job x[])
    {
        this.x = x;
    }
}

```

```

    Job nextJob()
    {
        return x[index ++];
    }
    boolean hasNext()
    {
        return index < x.length;
    }
}

class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
                        50000.9);
        Job j2 = new Job("hardware",
                        10000.9);
        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
    private static void print(Employee
emp)
    {
        Job j1 = null;

```

```

        while(emp.hasNext())
        {
            j1 = emp.nextJob();
            System.out.println(j1.title +
            ":" + j1.salary);
        }
    }
}

```

29. Program

```

package pack2;

class Job
{
    String title;
    double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}

class Employee
{
    private int index;
    Job x[];
    Employee(Job x[])
    {
        this.x = x;
    }
    Job nextJob()

```

```

    {
        return x[index ++];
    }
    boolean hasNext()
    {
        if(index == x.length)
        {
            index = 0;
            return false;
        }
        return true;
    }
}

class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
        50000.9);
        Job j2 = new Job("hardware",
        10000.9);
        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
}

```

```

        private static void print(Employee
emp)
    {
        Job j1 = null;
        while(emp.hasNext())
        {
            j1 = emp.nextJob();
            System.out.println(j1.title +
"." + j1.salary);
        }
    }
}

```

30. Program

```

package pack3;
class Job
{
    String title;
    double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}
class Employee
{
    Job x[];
    Employee(Job x[])
    {

```

```

        this.x = x;
    }
    Iterator getIterator()
    {
        Iterator it = new Iterator(x);
        return it;
    }
}
class Iterator
{
    private int index;
    Job x[];
    Iterator(Job x[])
    {
        this.x = x;
    }
    Job nextJob()
    {
        return x[index ++];
    }
    boolean hasNext()
    {
        return index < x.length;
    }
}
class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
50000.9);

```

```

        Job j2 = new Job("hardware",
10000.9);
        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
    private static void print(Employee
emp)
    {
        Iterator it = emp.getIterator();
        Job j1 = null;
        while(it.hasNext())
        {
            j1 = it.nextJob();

            System.out.println(j1.title + ":" +
j1.salary);
        }
    }
}

```

31. Program

```

package pack4;
class Job

```

```

{
    String title;
    double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}

class Employee
{
    Job x[];
    Employee(Job x[])
    {
        this.x = x;
    }
    Iterator getIterator()
    {
        Iterator it = new Iterator(x);
        return it;
    }
}

class Iterator
{
    private static int index;
    Job x[];
    Iterator(Job x[])
    {
        this.x = x;
    }
}

```

```

        Job nextJob()
        {
            return x[index ++];
        }
        boolean hasNext()
        {
            return index < x.length;
        }
    }
}
class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
50000.9);
        Job j2 = new Job("hardware",
10000.9);
        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
    private static void print(Employee
emp)
    {
        Iterator it = emp.getIterator();

```

```

        Job j1 = null;
        while(it.hasNext())
        {
            j1 = it.nextJob();
            System.out.println(j1.title +
":" + j1.salary);
        }
    }
}

```

32. Program

```

package pack5;
class Job
{
    String title;
    double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}
class Employee
{
    Job x[];
    Employee(Job x[])
    {
        this.x = x;
    }
    Iterator getIterator()

```

```

    {
        Iterator it = new Iterator(x);
        return it;
    }
    class Iterator
    {
        private int index;
        Job x[];
        Iterator(Job x[])
        {
            this.x = x;
        }
        Job nextJob()
        {
            return x[index ++];
        }
        boolean hasNext()
        {
            return index <
x.length;
        }
    }
} //end of employee
class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
50000.9);
        Job j2 = new Job("hardware",
10000.9);

```

```

        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
    private static void print(Employee
emp)
    {
        Employee.Iterator it
=emp.getIterator();
        Job j1 = null;
        while(it.hasNext())
        {
            j1 = it.nextJob();

            System.out.println(j1.title + ":" +
j1.salary);
        }
    }
}

```

33. Program

```

package pack6;
class Job
{

```

```

        String title;
        double salary;
    Job(String title, double salary)
    {
        this.title = title;
        this.salary = salary;
    }
}
class Employee
{
    Job x[];
    Employee(Job x[])
    {
        this.x = x;
    }
    Iterator getIterator()
    {
        Iterator it = new Iterator(x);
        return it;
    }
    static class Iterator
    {
        private static int index;
        Job x[];
        Iterator(Job x[])
        {
            this.x = x;
        }
    }
    Job nextJob()
    {
        return x[index ++];
    }
}

```

```

    }
    boolean hasNext()
    {
        return index <
            x.length;
    }
}
//end of employee
class Manager
{
    public static void main(String[] args)
    {
        Job j1 = new Job("software",
            50000.9);
        Job j2 = new Job("hardware",
            10000.9);
        Job j3 = new Job("hr", 500.9);
        Job x[] = {j1, j2, j3};
        Employee emp = new Employee(x);
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
        print(emp);
        System.out.println("----");
    }
    private static void print(Employee
emp)
    {
        Employee.Iterator it
=emp.getIterator();
    }
}

```



```

        Job j1 = null;
        while(it.hasNext())
        {
            j1 = it.nextJob();
            System.out.println(j1.title + ":" +
j1.salary);
        }
    }
}

```

34. Program

```

class A
{
    int i;
    static int j;
    void test1()
    {
        i = 20;
        j = 30;
        test1();
        test2();
    }
    static void test2()
    {
        //i = 20;
        j = 30;
        //test1();
        test2();
    }
}

```

35. Program

```

class A
{
    class B{}
    class C{}
}

```

36. Program

```

class B
{
    class C{}
    static class D{}
    void test1()
    {
        C c1 = new C();
        D d1 = new D();
    }
    static void test2()
    {
        //C c1 = new C();
        D d1 = new D();
    }
}

```

37. Program

```

class C
{
    int i;
    static int j;
    class D

```

```

    {
    }
    static class E
    {
    }
    void test1()
    {
        i = 1;
        j = 2;
        D d1 = new D();
        E e1 = new E();
        test1();
        test2();
    }
    static void test2()
    {
        i = 1;
        j = 2;
        D d1 = new D();
        E e1 = new E();
        test1();
        test2();
    }
}

```

38. Program

```

class D
{
    class E
    {

```

```

int i;
void test1()
{
}
//static int j;

/*
static void test2()
{
}
*/
}

```

39. Program

```

class E
{
    static class F
    {
        int i;
        static int j;
        void test1()
        {
        }
        static void test2()
        {
        }
    }
}

```

40. **Program**

class F

{

int i;

static int j;

void test1()

{

i = 1;

j = 2;

test1();

test2();

G g1 = new G();

H h1 = new H();

}

static void test2()

{

//i = 1;

j = 2;

//test1();

test2();

//G g1 = new G();

H h1 = new H();

}

class G

{

int m;

//static int n;

void test3()

{

i = 10;

j = 20;

test1();

test2();

G g1 = new G();

H h1 = new H();

m = 20;

//n = 20;

}

/*

static void test4()

{

i = 10;

j = 20;

test1();

test2();

G g1 = new G();

H h1 = new H();

m = 20;

n = 20;

}

*/

}

static class H

{

int p;

static int q;

void test5()

{

//i = 1;

j = 2;

//test1();

test2();

```

        p = 0;
        q = 9;
        //G g1 = new G();
        H h1 = new H();
    }
    static void test6()
    {
        //i = 10;
        j = 20;
        //test1();
        test2();
        //p = 9;
        q = 10;
        //G g1 = new G();
        H h1 = new H();
    }
}

```

41. Program

```

class G
{
    class H
    {
    }
    static class I
    {
    }
    public static void main(String[] args)

```

```

    {
        H h1 = null;
        I i1 = null;
        //h1 = new H();
        i1 = new I();
        System.out.println("done");
    }
}

```

42. Program

```

class H
{
    class I
    {
    }
    static class J
    {
    }
}
class Manager
{
    public static void main(String[] args)
    {
        H.I obj1 = null;
        H.J obj2 = null;
        obj1 = new H().new I();
        obj2 = new H.J();
        H h1 = new H();
        obj1 = h1.new I();
        System.out.println("done");
    }
}

```

}

43. Program

```

class I
{
    public static void main(String[] args)
    {
        class A
        {
            int i = 10;
        }
        A a1 = new A();
        System.out.println(a1.i);
        a1.i = 20;
        System.out.println(a1.i);
    }
}

```

44. Program

```

class J
{
    static void test()
    {
        class B
        {
            int i = 20;
        }
        B b1 = new B();
        System.out.println("b1.i:" + b1.i);
    }
}

```

```

        b1.i = 30;
        System.out.println("b1.i:" + b1.i);
    }
    public static void main(String[] args)
    {
        test();
        System.out.println("Hello World!");
    }
}

```

45. Program

```

class K
{
    public static void main(String[] args)
    {
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("----");
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("----");
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("1111");
    }
}

```

46. Program

```

class L
{
    static void test()
    {
        System.out.println("1111");
        System.out.println("1111");
        System.out.println("1111");
    }
    public static void main(String[] args)
    {
        test();
        System.out.println("----");
        test();
        System.out.println("----");
        test();
    }
}

```

47. Program

```

class M
{
    public static void main(String[] args)
    {
        class A
        {
            void test()
            {
                System.out.println("1111");
                System.out.println("1111");
            }
        }
    }
}

```

```

        System.out.println("1111");
    }
}
A a1 = new A();
a1.test();
System.out.println("----");
a1.test();
System.out.println("----");
a1.test();
}
}

```

48. Program

```

class N
{
    {
        class A
        {
        }
    }
}

N()
{
    class A
    {
    }
}

static
{
    class A

```

```

        {
        }
    }
    void test()
    {
        class A
        {
        }
    }
    public static void main(String[] args)
    {
        class A
        {
        }
        System.out.println("Hello World!");
    }
}

```

49. Program

```

class A
{
    A()
    {
        System.out.println(10);
    }
    public static void main(String[] args)
    {
        class B
        {
            B()
            {
                super();
                System.out.println("B()");
            }
        }
        B b1=new B();
    }
}

```

```

    }
}

```

50. Program

```

public class B
{
    public static int i=10;
    public static void main(String[] args)
    {
        private class C
        {
            private C()
            {
                System.out.println("local");
            }
        }
        C c1=new C();
        System.out.println("main");
    }
}

```

51. Program

```

public class B
{
    public static int i=10;
    public static void main(String[] args)
    {
        class C
        {
            private C()
            {
                System.out.println("local");
            }
        }
        C c1=new C();
        System.out.println("main");
    }
}

```

52. Program

```

class O
{
    public static void main(String[] args)
    {
        int i = 10;
        final int j = 20;
        class A
        {
            void test()
            {
                //System.out.println(i);
                System.out.println(j);
            }
            System.out.println("done");
        }
    }
}

```

53. Program

```

class P
{
    public static void main(String[] args)
    {
        class A
        {
            static int i;
        }
        System.out.println("done");
    }
}

```

54. Program

```

class A
{
    void test1()
    {
        System.out.println("A-
test1");
    }
    void test2()
    {
        System.out.println("A-
test2");
    }
}

```

55. Program

```

abstract class B
{
    abstract void test1();
    void test2()
    {
        System.out.println("B-
test2");
    }
}

```


56. Program

```
interface C
{
    void test1();
    void test2();
}
```

57. Program

```
class Manager1
{
    public static void main(String[] args)
    {
        A a1 = new A();
        a1.test1();
        a1.test2();
        System.out.println("done");
    }
}
```

58. Program

```
class Manager2
{
    public static void main(String[] args)
    {
        A a1 = new A()
        {
            void test1()
            {
```

```
        System.out.println("AIC-
test1");
    }
};
a1.test1();
a1.test2();
System.out.println("Hello World!");
}
```

59. Program

```
class Manager3
{
    public static void main(String[] args)
    {
        A a1 = new A();
        A a2 = new A()
        {
            void test2()
            {
                System.out.println("AIC-
test2");
            }
        };
        a1.test1();
        a1.test2();
        a2.test1();
        a2.test2();
        System.out.println("done");
    }
}
```

```
    }
}
```

60. Program

```
class Manager4
{
    public static void main(String[] args)
    {
        A a1 = new A()
        {
        };
        a1.test1();
        a1.test2();
        System.out.println("-----");
        A a2 = new A()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
        };
        a2.test1();
        a2.test2();
        System.out.println("-----");
        A a3 = new A()
        {
            void test2()
            {
```

```
                System.out.println("AIC-
test2");
            }
        };
        a3.test1();
        a3.test2();
        System.out.println("-----");
        A a4 = new A()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
            void test2()
            {
                System.out.println("AIC-
test2");
            }
        };
        a4.test1();
        a4.test2();
    }
}
```

61. Program

```
class Manager5
{
    public static void main(String[] args)
    {
        B b1 = new B();
```

```

        b1.test1();
        b1.test2();
        System.out.println("done");
    }
}

```

62. Program

```

class Manager6
{
    public static void main(String[] args)
    {
        B b1 = new B()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
        };
        b1.test1();
        b1.test2();
        System.out.println("done");
    }
}

```

63. Program

```

class Manager7
{
    public static void main(String[] args)
    {

```

```

        B b1 = new B()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
            void test2()
            {
                System.out.println("AIC-
test2");
            }
        };
        b1.test1();
        b1.test2();
        System.out.println("done");
    }
}

```

64. Program

```

class Manager8
{
    public static void main(String[] args)
    {
        C c1 = new C();
        c1.test1();
        c1.test2();
        System.out.println("done");
    }
}

```

65. Program

```

class Manager9
{
    public static void main(String[] args)
    {
        C c1 = new C()
        {
            public void test1()
            {
                System.out.println("AIC-
test1");
            }
            public void test2()
            {
                System.out.println("AIC-
test2");
            }
        };
        c1.test1();
        c1.test2();
        System.out.println("done");
    }
}

```

66. Program

```

class Manager10
{
    static void method1(B b1)
    {
        b1.test1();
    }
}

```

```

        b1.test2();
    }
    static void method2(C c1)
    {
        c1.test1();
        c1.test2();
    }
    public static void main(String[] args)
    {
        System.out.println("done");
    }
}

```

67. Program

```

class Manager11
{
    static void method1(B b1)
    {
        b1.test1();
        b1.test2();
    }
    static void method2(C c1)
    {
        c1.test1();
        c1.test2();
    }
    public static void main(String[] args)
    {
        B b1 = null;
        method1(b1);
    }
}

```

```

        C c1 = null;
        method2(c1);
        System.out.println("done");
    }
}

```

68. Program

```

class Manager12
{
    static void method1(B b1)
    {
        b1.test1();
        b1.test2();
    }
    static void method2(C c1)
    {
        c1.test1();
        c1.test2();
    }
    public static void main(String[] args)
    {
        B b1 = new B()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
        }
    }
}

```

```

        };
        method1(b1);
        C c1 = new C()
        {
            public void test1()
            {
                System.out.println("AIC-
test1");
            }
            public void test2()
            {
                System.out.println("AIC-test2");
            }
        };
        method2(c1);
        method1(new B()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
        });
        method2(new C()
        {
            public void test1()
            {
                System.out.println("AIC-
test1");
            }
        }
    }
}

```

```

        public void test2()
        {
            System.out.println("AIC-
test2");
        }
    });
    System.out.println("done");
}
}

```

69. Program

class A

```

{
    public static void main(String[] args)
    {
        System.out.println(1);
        abstract class B
        {
            abstract void test();
        }
        public void main(String[] array)
        {
            System.out.println(3);
        }
        B b1 = new B()
        {
            void test()
            {
                System.out.println(2);
            }
        };
        b1.test();
        b1.main(args);
    }
}
}

```

70. Program

class Manager13

```

{
    public static void main(String[] args)
    {
        B b1 = method1();
        C c1 = method2();
        b1.test1();
        b1.test2();
        c1.test1();
        c1.test2();
        System.out.println("done");
    }
    static B method1()
    {
        B b1 = new B()
        {
            void test1()
            {
                System.out.println("AIC-
test1");
            }
        };
        return b1;
    }
    static C method2()
    {
        return new C()
        {
            public void test1()
            {
                System.out.println("AIC-
test1");
            }
        }
    }
}

```

```

        }
        public void test2()
        {
            System.out.println("AIC-
test2");
        }
    }
}

```

71. Program

```

interface Switch
{
    void on();
    void off();
}
class Fan
{
    private boolean runningStatus;
    public Switch getSwitch()
    {
        return new Switch()
        {
            public void on()
            {
                runningStatus = true;
            }
            public void off()
            {
                runningStatus = false;
            }
        }
    }
}

```

```

        };
    }
    public boolean getRunningStatus()
    {
        return runningStatus;
    }
}
class Manager14
{
    public static void main(String[] args)
    {
        Fan f1 = new Fan();
        Switch s1 = f1.getSwitch();
        s1.on();
        System.out.println(f1.getRunningSta
tus());
        s1.off();
        System.out.println(f1.getRunningSta
tus());
    }
}

```

72. Program

```

class D
{
    D()
    {
        System.out.println("D()");
    }
    D(int i)
    {

```

```

        System.out.println("D(int)");
    }

    {
        System.out.println("D-IIB");
    }
}
class Manager15
{
    public static void main(String[] args)
    {
        D d1 = new D()
        {
            {
                System.out.println("AIC-
IIB");
            }
        };
        System.out.println("-----");
        D d2 = new D(10)
        {
            {
                System.out.println("AIC-
IIB");
            }
        };
    }
}

```

73. Program

```

class A
{
    void test1()
    {
        System.out.println("A-
test1");
    }
}

```

74. Program

```

class B extends A
{
    @Override
    void test1()
    {
        System.out.println("B-
test1");
    }

    void test2()
    {
        System.out.println("B-
test2");
    }
}

```

75. Program

```

class C
{
    @Deprecated

```



```

void test1()
{
    //some content
}

void test2()
{
    //again some content
} }

```

76. Program

```

class Manager1
{
    public static void main(String[] args)
    {
        C c1 = new C();
        c1.test1();
        c1.test2();
        System.out.println("done");
    }
}

```

77. Program

```

class Manager2
{
    public static void main(String[] args)
    {
        Thread t1 = new Thread();
        t1.stop();
        System.out.println("done");
    }
}

```

```

}

```

78. Program

```

@SuppressWarnings(value="deprecation")
class Manager3
{
    public static void main(String[] args)
    {
        Thread t1 = new Thread();
        t1.stop();
        C c1 = new C();
        c1.test1();

        System.out.println("Hello World!");
    }
}

```

79. Program

```

import java.util.ArrayList;
class Manager4
{
    @SuppressWarnings(value="unchecked")
    public static void main(String[] args)
    {
        ArrayList list = new
        ArrayList();
        list.add(90);
        list.add(90);
        System.out.println("done");
    }
}

```

```
}

```

80. Program

```
import java.util.ArrayList;
class Manager5
{
    @SuppressWarnings({"unchecked",
"deprecation"})
    public static void main(String[] args)
    {
        ArrayList list = new
ArrayList();
        list.add(90);
        list.add(90);

        Thread t1 = new Thread();
        t1.stop();

        C c1 = new C();
        c1.test1();
        System.out.println("Hello
World!");
    }
}
```

81. build.xml

```
<project default="run">
    <target name="setup">
        <echo message="setup is
starting"/>
        <mkdir dir="src"/>
        <mkdir dir="classes"/>
        <mkdir dir="lib"/>
        <mkdir dir="docs"/>
    </target>
    <target name="cleanup">
        <echo message="cleaning
started"/>
        <delete dir="classes"/>
        <mkdir dir="classes"/>
        <delete dir="lib"/>
        <mkdir dir="lib"/>
        <delete dir="docs"/>
        <mkdir dir="docs"/>
    </target>
    <target name="compile"
depends="cleanup">
        <echo message="compilation
started"/>
        <javac srcdir="src"
destdir="classes"/>
    </target>

```

```

        <target name="jar"
depends="compile">
            <echo message="making a
jar started"/>
            <jar basedir="classes"

destfile="lib/mgr.jar"/>
        </target>
        <target name="doc-creation"
depends="jar">
            <javadoc destdir="docs"
                sourcepath="src"
                packagenames="com.lara"/>
        </target>
        <target name="run" depends="doc-
creation">
            <java classpath="classes"

classname="com.lara.Manager"/>
        </target>
</project>

```

82. Program

```

package com.lara;

public class I
{
    public static void main(String[] args)
    {
        int i = 0;
        i++;
    }
}

```

```

        ++i;
        i += 2;
        i -= 3;
        i++;
        ++i;
        System.out.println(i);
    }
}

```

83. Program

```

class Manager
{
    public static void main(String[] args)
    {
        int[]x = {10, 20, 30, 200};
        for (int i : x)
        {
            System.out.println(i);
        }
    }
}

```

84. Program

```

class Manager1
{
    public static void main(String[] args)
    {
        int x[] = {10, 2, 5, 90, 20};
        for (int i = 0; i < x.length; i++)
        {

```

```

        System.out.println(i);
    }
    for ( int i : x)
    {
        System.out.println(i);
    }
}
}

```

85. Program

```

class Manager2
{
    public static void main(String[] args)
    {
        String[] x = {"abc",
                      "xyz",
                      "hello",
                      "test",
                      "done"
        };
        for (String s1 : x)
        {
            System.out.println(s1);
        }
    }
}
}

```

86. Program

```

package com.lara;
public class M1

```

```

{
    public static void main(String[] args)
    {
        int i=0;
        Integer obj=new Integer(i);
        int k=obj.intValue();
        System.out.println("done");
    }
}
}

```

87. Program

```

package com.lara;
public class M2
{
    public static void main(String[] args)
    {
        Double d1=new Double(10.09);
        double d2=d1.doubleValue();
        System.out.println("done");
    }
}
}

```

88. Program

```

package com.lara;
public class M3
{
    public static void main(String[] args)
    {
        char c1='a';
        Character c2=new Character('a');
    }
}

```

```

        char c3=c2.charValue();
        System.out.println();
    }
}

```

89. Program

```

package com.lara;

public class M4
{
    public static void main(String[] args)
    {
        String s1="10";
        Integer obj=new Integer(s1);
        int i=obj.intValue();
        Double obj2=new Double(s1);
        double d=obj2.doubleValue();
        Long obj3=new Long(s1);
        Long l1=obj3.longValue();
        System.out.println("done");
    }
}

```

90. Program

```

package com.lara;

public class M5
{
    public static void main(String[] args)
    {
        String s1="true";
        String s2="abc";
    }
}

```

```

        Boolean b1=new Boolean(s1);
        Boolean b2=new Boolean(s2);
        System.out.println(b1);
        System.out.println(b2);
    }
}

```

91. Program

```

package com.lara;

public class M6
{
    public static void main(String[] args)
    {
        String s1="abc";
        Boolean obj=new Boolean(s1);
        System.out.println(obj);
    }
}

```

92. Program

```

package com.lara;

public class M7
{
    public static void main(String[] args)
    {
        String s1="abc";
        Integer obj=new Integer(s1);
        System.out.println(obj);
    }
}

```

93. Program

```

package com.lara;

public class M8
{
    public static void main(String[] args)
    {
        String s1="10";

        int i=Integer.parseInt(s1);
        double d=Double.parseDouble(s1);
        float f1=Float.parseFloat(s1);
        byte b1=Byte.parseByte(s1);
        System.out.println("done");
    }
}

```

94. Program

```

package com.lara;

public class M9
{
    public static void main(String[] args)
    {
        int i=10;
        double d=10.09;
        byte b=10;

        String s1=Integer.toString(i);
        String s2=Double.toString(d);
        String s3=Byte.toString(b);
        System.out.println("done");
    }
}

```

```

}
}

```

95. Program

```

package com.lara;

public class M10
{
    public static void main(String[] args)
    {
        int i=10;
        String s1="10";
        Integer obj1=Integer.valueOf(s1);
        Integer obj2=Integer.valueOf(i);
        System.out.println("done");
    }
}

```

96. Program

```

package com.lara;

public class M11
{
    public static void main(String[] args)
    {
        double d1=10.09;
        String s1="20.90";
        Double d2=Double.valueOf(d1);
        Double d3=Double.valueOf(s1);
        System.out.println("done");
    }
}

```

97. Program

```

package com.lara;

public class M12
{
    public static void main(String[] args)
    {
        Integer obj=10;
        int i=obj;
        System.out.println("done");
    }
}

```

98. Program

```

//Jdk 1.4
package com.lara;
public class M13
{
    public static void main(String[] args)
    {
        Integer obj=10;
        int i=obj;
        System.out.println("done");
    }
}

```

99. Program

```

//Jdk 1.4
package com.lara;
public class M14
{

```

```

    public static void main(String[] args)
    {
        Integer obj=new Integer(10);
        int i=obj.intValue();
        System.out.println("done");
    }
}

```

100. Program

```

//Jdk 1.6
package com.lara;
public class M15
{
    public static void main(String[] args)
    {
        Integer obj=new Integer(10);
        test(obj);
    }
    static void test(int i)
    {
        System.out.println("done");
    }
}

```

101. Program

```

//JDK1.6
package com.lara;
public class M16
{
    public static void main(String[] args)
    {
        Integer obj=new Integer(10);

```

```

        test(obj.intValue());
    }
    static void test(int i)
    {
        System.out.println("done");
    }
}

```

102. Program

```

class Test
{
    public static void main(String args[])
    {
        Integer i1=4678;
        Integer i2=4678;
        if(i1==i2)
        {
            System.out.println("same Objects");
        }
        else
        {
            System.out.println("Different Objects");
        }
        if(i1.equals(i2))
        {
            System.out.println("Meaningfully equal");
        }
        else
        {
            System.out.println("not equal");
        }
    }
}

```

103. Program

```

//JDK1.6
package com.lara;
public class M17
{

```

```

    public static void main(String[] args)
    {
        int i=test();
    }
    static Integer test()
    {
        return 20;
    }
}

```

104. Program

```

//Jdk1.4
package com.lara;
public class M18
{
    public static void main(String[] args)
    {
        int i=test();
    }
    static Integer test()
    {
        return 20;
    } }

```

105. Program

```

//JDK1.4
package com.lara;
public class M19
{
    public static void main(String[] args)
    {

```



```

        int i=test().intValue();

    }
    static Integer test()
    {
        return new Integer(20);
    }
}

```

106. Program

```

//jdk1.6
package com.lara;
public class M20
{
    public static void main(String[] args)
    {
        Boolean flag=true;
        if(flag)
        {
            System.out.println(flag);
        }
        else
        {
            System.out.println(flag);
        }
    }
}

```

107. Program

```

//JDK1.6
package com.lara;

```

```

public class M21
{
    public static void main(String[] args)
    {
        Boolean flag=true;
        if(flag)
        {
            System.out.println(flag);
        }
        else
        {
            System.out.println(flag);
        }
    }
}

```

108. Program

```

//JDK1.6
package com.lara;
public class M21
{
    public static void main(String[] args)
    {
        Boolean flag=true;
        if(flag)
        {
            System.out.println(flag);
        }
        else
        {

```

```

        System.out.println(flag);
    }
}

```

109. Program

```

//Jdk1.4
package com.lara;
public class M23
{
    public static void main(String[] args)
    {
        char c1=new Character('a');
        Character c2=c1;
        System.out.println("done");
    }
}

```

110. Program

```

//JDK1.4
package com.lara;
public class M24
{
    public static void main(String[] args)
    {
        char c1=new
        Character('a').charValue();
        Character c2=new Character(c1);
        System.out.println("done");
    }
}

```

111. Program

```

package com.lara;
public class M25
{
    static void test(int i)
    {
        System.out.println("int");
    }
    static void test(Integer i)
    {
        System.out.println("Integer");
    }
    public static void main(String[] args)
    {
        int i = 10;
        test(i);
        Integer obj = new Integer(90);
        test(obj);
    }
}

```

112. Program

```

package com.lara;
public class M26
{
    static void test(int i)
    {
        System.out.println("int");
    }
    static void test(Double d)
    {

```

```

        System.out.println("Double");
    }
    public static void main(String[] args)
    {
        Integer obj = new Integer(90);
        test(obj);
        double d1 = 90.90;
        test(d1);
    }
}

```

113. Program

```

package com.lara;
public class M27
{
    static void test(byte b)
    {
        System.out.println("byte");
    }
    static void test(short b)
    {
        System.out.println("short");
    }
    static void test(int b)
    {
        System.out.println("int");
    }
    static void test(long b)
    {
        System.out.println("long");
    }
}

```

```

    }
    static void test(float b)
    {
        System.out.println("float");
    }
    static void test(double b)
    {
        System.out.println("double");
    }
    static void test(Byte b)
    {
        System.out.println("Byte");
    }
    static void test(Integer b)
    {
        System.out.println("Integer");
    }
    static void test(Double b)
    {
        System.out.println("Double");
    }
    static void test(Number b)
    {
        System.out.println("Number");
    }
    static void test(Object b)
    {
        System.out.println("Object");
    }
}

```

```

static void test(byte ... b)
{
    System.out.println("byte ...");
}

public static void main(String[] args)
{
    byte b = 10;
    test(b);
}
}

```

114. Program

```

class W7
{
    static void test(Byte b,Byte i)
    {
        System.out.println("Byte");
    }
    static void test(byte ... b)
    {
        System.out.println("byte");
    }
    public static void main (String args[])
    {
        byte b=10;
        test(b,b);
    }
}

```

115. Program

```

class W8
{
    static void test(Byte b,Byte b)
    {
        System.out.println("Byte");
    }
}

```

```

static void test(byte ... b)
{
    System.out.println("byte");
}

public static void main (String args[])
{
    byte b=10;
    test(b,b);
}
}

```

116. Program

```

package com.lara;
public class M28
{
    static void test()
    {
        System.out.println("test()");
    }
    static void test(int ... x)
    {
        System.out.println("int ... ");
    }
    public static void main(String[] args)
    {
        test();
        test(10);
        test(20, 30);
        test(20, 30, 3, 50, 60);
    }
}

```

117. Program

```

package com.lara;
public class M29

```

```

{
    static void test(int ... x)
    {
        System.out.println(x.length);
        for(int i : x)
        {
            System.out.println(i);
        }
        System.out.println("-----");
    }
    public static void main(String[] args)
    {
        test();
        test(20, 30);
        test(90, 9, 20, 40);
        test(80, 2, 34, 450, 1000);
    }
}

```

118. Program

```

package com.lara;
public class M30
{
    static void test(int ... x)
    {
        System.out.println(x.length);
        for(int i : x)
        {
            System.out.println(i);
        }
        System.out.println("-----");
    }
}

```

```

public static void main(String[] args)
{
    test();
    test(20, 30);
    test(90, 9, 20, 40);
    test(80, 2, 34, 450, 1000);
}

```

119. Program

```

package com.lara;
public class M31
{
    static void test(String ... strings)
    {
        for(String str : strings)
        {
            System.out.print(str + ",");
        }
        System.out.println();
    }
    public static void main(String[] args)
    {
        test("msg1");
        test("msg1", "msg2");
        test("msg1", "msg2", "msg3");
        test("msg1", "msg2", "msg3",
            "msg4");
    }
}

```

120. Program

```

package com.lara;
public class M32
{
    static void test1(String s1, int ... i)
    {

    }

    static void test2(String ... s1, byte b)
    {

    }

}

```

121. Program

```

package com.lara;
public class M33
{
    static void test(int ... x)
    {

    }

    static void test(int[] y)
    {

    }

}

```

122. Program

```

package com.lara;
public class M34

```

```

{
    public static void main(String ... x)
    {
        System.out.println("Hello to all");
    }

}

```

123. Program

```

class A
{
    public static void main(String[] args)
    {
        System.out.println("Hello
World!");
    }

}

```

124. Program

```

class B
{
    public static void main(String[] args)
    {
        System.out.println("Hello " +
args[0]);
    }

}

```

125. Program

```

class C
{
    public static void main(String[] args)
    {
        System.out.println(args.length);
    }

}

```

}

126. Program

```

class D
{
    public static void main(String[] args)
    {
        for(String s1 : args)
        {
            System.out.println(s1);
        }
    }
}

```

127. Program

```

package com.lara;
import java.util.Scanner;
public class Manager
{
    public static void main(String[] args)
    {
        Scanner sc = new
        Scanner(System.in);
        System.out.println("enter
        some thing");
        String s1 = sc.next();
        System.out.println("you have
        entered:" + s1);
    }
}

```

128. Program

```

package com.lara;
import java.util.Scanner;

```

```

public class Manager1
{
    public static void main(String[] args)
    {
        Scanner sc =
            new
        Scanner(System.in);
        System.out.println("enter
        string value");
        String s1 = sc.next();
        System.out.println("enter int
        value");
        int i = sc.nextInt();
        System.out.println("enter
        double value");
        double j = sc.nextDouble();
        System.out.println("enter boolean
        value");
        boolean b = sc.nextBoolean();
        System.out.println("you have
        entered");
        System.out.println(s1);
        System.out.println(i);
        System.out.println(j);
        System.out.println(b);
        System.out.println("done");
    }
}

```

129. Program

```

package com.lara;

```

```

public class A
{
    public static void main(String[] args)
    {
        System.out.println(1);
        int i = 10/0;
        System.out.println(2);
    }
}

```

130. Program

```

package com.lara;
public class B
{
    public static void main(String[] args)
    {
        System.out.println(1);
        test();
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        int i = 10/0;
        System.out.println(4);
    }
}

```

131. Program

```

package com.lara;
public class C
{
    public static void main(String[] args)
    {
        System.out.println(1);
        test1();
        System.out.println(2);
    }
    static void test1()
    {
        System.out.println(3);
        test2();
        System.out.println(4);
    }
    static void test2()
    {
        System.out.println(5);
        String s1 = null;

        System.out.println(s1.length());
        System.out.println(6);
    }
}

```

132. Program

```

package com.lara;
public class D
{
    static void test1()

```



```

        {
            System.out.println(1);
            test2();
            System.out.println(2);
        }
        static void test2()
        {
            System.out.println(3);
            int i =
Integer.parseInt("abc");
            System.out.println(4);
        }
        public static void main(String[] args)
        {
            System.out.println(5);
            test1();
            System.out.println(6);
        }
    }

```

133. Program

```

package com.lara;
public class E
{
    static void print()
    {
        System.out.println(1);
        int i = 10/0;
        System.out.println(2);
    }
}

```

```

    }

```

134. Program

```

package com.lara;
public class F
{
    public static void main(String[] args)
    {
        System.out.println(100);
        E.print();
        System.out.println(200);
    }
}

```

135. Program

```

package com.lara;
public class G
{
    public static void main(String[] args)
    {
        System.out.println(1);
        main(args);
        System.out.println(2);
    }
}

```

136. Program

```

package com.lara;
public class H

```

```

{
    public static void main(String[] args)
    {
        System.out.println(1);
        test();
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        test();
        System.out.println(4);
    }
}

```

137. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.println(1);
        int x[] = new int[999999999];
        System.out.println(2);
    }
}

```

138. Program

```

package com.lara;
public class J
{

```

```

public static void main(String[] args)
{
    System.out.println(1);
    try
    {
        int i = 10/0;
    }
    catch(ArithmeticException
ex)
    {
        System.out.println(2);
        System.out.println(ex);
        System.out.println(3);
    }
    System.out.println(4);
}
}

```

139. Program

```

package com.lara;
public class K
{
    public static void main(String[] args)
    {
        System.out.println(1);

        try
        {
            System.out.println(2);
            int i = 10/0;

```

```

        System.out.println(3);
    }
    catch(ArithmeticException
ex)
    {
        System.out.println(4);
        System.out.println(ex.getMessage());
        System.out.println(5);
    }
    System.out.println(6);
}

```

Program

```

package com.lara;
public class L
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);
            int i = 10/0;
            System.out.println(2);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(3);
            ex.printStackTrace();
            System.out.println(4);

```

```

    }
    System.out.println(5);
}

```

140. Program

```

package com.lara;
public class M
{
    public static void main(String[] args)
    {
        try
        {
            int i = 10;
            System.out.println(i);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(ex);

            //System.out.println(i);
            int j = 20;
            System.out.println(j);
        }
        //System.out.println(i);
        //System.out.println(j);
    }
}

```

141. Program

```

package com.lara;

public class N
{
    public static void main(String[] args)
    {
        int i = 0;
        try
        {
            System.out.println(i);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(i);
        }
        System.out.println(i);
    }
}

```

142. Program

```

package com.lara;

public class O
{
    public static void main(String[] args)
    {
        System.out.println(1);
        int i = 10/0;
    }
}

```

```

try
{
    System.out.println(2);
}
catch(ArithmeticException
ex)
{
    System.out.println(3);
}
System.out.println(4);
}
}

```

143. Program

```

package com.lara;

public class P
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try
        {
            System.out.println(2);
            int i = 10/0;
            System.out.println(3);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(4);
        }
    }
}

```

```

        int j = 20/0;
        System.out.println(5);
    }
    System.out.println(6);
}

```

144. Program

```

package com.lara;
public class Q
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try
        {
            System.out.println(2);
            int i = 10/0;
            System.out.println(3);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(4);
            try
            {
                System.out.println(5);
                int i = 20/0;
                System.out.println(6);
            }

```

```

        catch(ArithmeticException ex1)
        {
            System.out.println(7);
        }
        System.out.println(8);
    }
    System.out.println(9);
}

```

145. Program

```

package com.lara;
import java.util.Scanner;
public class R
{
    public static void main(String[] args)
    {
        Scanner sc = new
Scanner(System.in);
        System.out.println("enter some
thing");
        String s1 = sc.next();
        try
        {
            System.out.println(1);
            int i = Integer.parseInt(s1);
            System.out.println(2);
            int k = i / (i - 9);
            System.out.println(3);

```

```

    }

    catch(NumberFormatException ex)
    {
        System.out.println(4);

        System.out.println(ex);
        System.out.println(5);
    }
    System.out.println(6);
}
}

```

146. Program

```

package com.lara;
import java.util.Scanner;
public class S
{
    public static void main(String[] args)
    {
        Scanner sc = new
Scanner(System.in);
        System.out.println("enter some
thing");
        String s1 = sc.next();
        try
        {
            System.out.println(1);
            int i = Integer.parseInt(s1);
            System.out.println(2);
            int k = i / (i - 9);

```

```

        System.out.println(3);
    }
    catch(ArithmeticException
ex)
    {
        System.out.println(4);

        System.out.println(ex);
        System.out.println(5);
    }
    catch(NumberFormatException ex)
    {
        System.out.println(6);

        System.out.println(ex);
        System.out.println(7);
    }
    System.out.println(8);
}
}

```

147. Program

```

package com.lara;
import java.util.Scanner;
public class T
{
    public static void main(String[] args)
    {

```

```

        Scanner sc = new
Scanner(System.in);
        System.out.println("enter some
thing");
        String s1 = sc.next();
        try
        {
            System.out.println(1);
            int i = Integer.parseInt(s1);
            System.out.println(2);
            int k = i / (i - 9);
            System.out.println(3);
        }
        catch(NumberFormatException ex)
        {
            System.out.println(4);
            System.out.println(ex);
            System.out.println(5);
        }
        finally
        {
            System.out.println("finally");
        }
        System.out.println(6);
    }
}

```

148. Program

```

package com.lara;
import java.util.Scanner;
public class U
{
    public static void main(String[] args)
    {
        Scanner sc = new
Scanner(System.in);
        System.out.println("enter some
thing");
        String s1 = sc.next();
        int i = test(s1);
        System.out.println(i);
    }
    static int test(String s1)
    {
        int i = 0;
        try
        {
            i =
Integer.parseInt(s1);
            return i;
        }
        catch(NumberFormatException ex)
        {
            return 0;
        }
        finally
        {
            return 1000;
        }
    }
}

```

```

    }
}

```

149. Program

```

package com.lara;
public class V
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);
            return;
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(2);
        }
        finally
        {
            System.out.println(3);
        }
        System.out.println(4);
    }
}

```

150. Program

```

package com.lara;
public class W
{
    public static void main(String[] args)
    {
        if(true)
        {
            return;
        }
        try
        {
            System.out.println(1);
            return;
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(2);
        }
        finally
        {
            System.out.println(3);
        }
        System.out.println(4);
    }
}

```

151. Program

```

package com.lara;
public class X

```



```

{
    public static void main(String[] args)
    {
        System.out.println(1);
        int i =
Integer.parseInt("abc");
        System.out.println(2);
        try
        {
            System.out.println(3);
        }
        catch(NullPointerException
ex)
        {
            System.out.println(4);
        }
        finally
        {
            System.out.println(5);
        }
        System.out.println(6);
    } }

```

152. Program

```

package com.lara;
public class Y
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try

```

```

{
    System.out.println(1);
    int i = 10/0;
    System.out.println(2);
}
catch(ArithmeticException
ex)
{
    System.out.println(3);
    int i = 20/0;
    System.out.println(4);
}
finally
{
    System.out.println(5);
}
System.out.println(6);
}
}

```

153. Program

```

package com.lara;
public class Z
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try

```

```

        {
            System.out.println(2);
            System.exit(0);
            System.out.println(3);
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(4);
        }
        finally
        {
            System.out.println(5);
        }
        System.out.println(6);
    }
}

```

154. Program

```

package com.lara;
public class A
{

```

```

    int test(boolean flag)
    {
        return 1000;
    }
    int test1(boolean flag)
    {
        if(flag)
        {

```

```

            return 10;
        }
        else
        {
            return 20;
        }
    }

    // int test3(boolean flag)
    // {
    //     if(flag)
    //     {
    //         return 10;
    //     }
    // }

    int test4(boolean flag)
    {
        if(flag)
        {
            return 10;
        }
        return 20;
    }

    // int test5(boolean flag)
    // {
    //     if(flag)
    //     {
    //     }
    //     else
    //     {

```

```
//          return 20;
//      }
//  }
int test6(boolean flag)
{
    if(flag)
    {

    }
    else
    {
        return 20;
    }
    return 0;
}

// int test7(boolean flag)
// {
//     if(flag)
//     {
//         return 200;
//     }
//     else
//     {
//         return 20;
//     }
//     return 0;
// }

}
```

155. Program

```
package com.lara;

public class B
{
    int test(String s1)
    {
        try
        {
            //some stmts
        }
        catch(NumberFormatException ex)
        {
        }
        return 20;
    }

    int test1(String s1)
    {
        try
        {
            //some stmts
            return 1;
        }
        catch(NumberFormatException ex)
        {
            return 0;
        }
    }
}
```

```
//      int test2(String s1)
//      {
//          try
//          {
//              //some stmts
//          }
//      }
//
//      catch(NumberFormatException ex)
//      {
//          return 0;
//      }
//  }
```

```
int test3(String s1)
{
    try
    {
        //some stmts
    }

    catch(NumberFormatException ex)
    {
        return 0;
    }

    return 500;
}

//      int test4(String s1)
//      {
```

```
//          try
//          {
//              //some stmts
//              return 20;
//          }

//      catch(NumberFormatException ex)
//      {
//          }
//      }

//      int test5(String s1)
//      {
//          try
//          {
//              //some stmts
//              return 20;
//          }

//      catch(NumberFormatException ex)
//      {
//          }

//          return 30;
//      }

//      int test6(String s1)
//      {
//          try
//          {
```

```
//          //some stmts
//          return 20;
//      }
//
//      catch(NumberFormatException ex)
//      {
//          return 0;
//      }
//      return 30;
//  }

}
```

156. Program

```
package com.lara;
public class C
{
    int test1()
    {
        try
        {
            //some stmts
        }
        catch(NullPointerException
ex)
        {
        }
        catch(ArithmeticException
ex)
        {
        }
    }
}
```

```
    }
    return 10;
}

int test2()
{
    try
    {
        //some stmts
        return 10;
    }
    catch(NullPointerException
ex)
    {
        return 20;
    }
    catch(ArithmeticException
ex)
    {
        return 30;
    }
}

// int test3()
// {
//     try
//     {
//         //some stmts
//     }
//     catch(NullPointerException
ex)
}
```

```
//      {
//          return 20;
//      }
//      catch(ArithmeticException
ex)
//      {
//          return 30;
//      }
//  }
```

```
int test4()
{
    try
    {
        //some stmts

    }
    catch(NullPointerException
ex)
    {
        return 20;
    }
    catch(ArithmeticException
ex)
    {
        return 30;
    }
    return 300;
}
```

```
//  int test5()
//  {
```

```
//      try
//      {
//          //some stmts

//      }
//      catch(NullPointerException
ex)
//      {

//      }
//      catch(ArithmeticException
ex)
//      {
//          return 30;
//      }
//  }
```

```
int test6()
{
    try
    {
        //some stmts

    }
    catch(NullPointerException
ex)
    {

    }
    catch(ArithmeticException
ex)
    {
```

```

        return 30;
    }
    return 1000;
}

// int test7()
// {
//     try
//     {
//         //some stmts
//         return 30;
//     }
//     catch(NullPointerException
ex)
//     {
//         return 40;
//     }
//     catch(ArithmeticException
ex)
//     {
//         return 50;
//     }
//     return 70;
// }

}

return 30;
}
return 1000;
}

// int test7()
// {
//     try
//     {
//         //some stmts
//         return 30;
//     }
//     catch(NullPointerException
ex)
//     {
//         return 40;
//     }
//     catch(ArithmeticException
ex)
//     {
//         return 50;
//     }
//     return 70;
// }

}

{
    int test1()
    {
        try
        {
            //some stmts
        }
        catch(ArithmeticException
ex)
        {
        }
        finally
        {
        }
        return 10;
    }

    int test2()
    {
        try
        {
            //some stmts
        }
        catch(ArithmeticException
ex)
        {
        }
        finally
        {
        }
    }
}

```

157. Program

```

package com.lara;
public class D

```

```

        return 30;
    }
}

// int test3()
// {
//     try
//     {
//         //some stmts
//     }
//     catch(ArithmeticException
// ex)
//     {
//     }
//     finally
//     {
//         return 30;
//     }
//     return 40;
// }

int test4()
{
    try
    {
        //some stmts
    }
    catch(ArithmeticException
ex)
    {
        return 40;
    }
}

```

```

        finally
        {
        }
    }
    return 40;
}

// int test5()
// {
//     try
//     {
//         //some stmts
//         return 20;
//     }
//     catch(ArithmeticException
// ex)
//     {
//         return 30;
//     }
//     finally
//     {
//     }
//     return 40;
// }

// int test6()
// {
//     try
//     {
//         //some stmts
//         return 20;
//     }
// }

```



```
//      catch(ArithmeticException
ex)
//      {
//          return 30;
//      }
//      finally
//      {
//          return 40;
//      }
//      return 50;
//  }
}
```

158. Program

```
package com.lara;
import java.io.FileWriter;
import java.io.IOException;
import java.lang.instrument.ClassDefinition;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.ParseException;
public class E
{
    void test1()
    {
        System.out.println(1);
        int i = 10/0;
        System.out.println(2);
    }
}
```

```
void test2()
{
    System.out.println(1);
    try
    {
    }
    catch(ArithmeticException
ex)
    {
    }
    System.out.println(2);
}

void test3()
{
    System.out.println(1);
    int i =
Integer.parseInt("abc");
    System.out.println(2);
}

void test4()
{
    System.out.println(1);
    try
    {
    }
    catch(NumberFormatException ex)
    {
    }
}
```

```

System.out.println(ex);
    }
    System.out.println(2);
}
void test5()
{
    System.out.println(1);
    String s1 = null;
    s1.length();
    System.out.println(2);
}
void test6()
{
    System.out.println(1);
    try
    {
        ex)
    }
    catch(NullPointerException
    {
        ex)
    }
    System.out.println(2);
}
void test7()
{
    Object obj = new Object();
    E e1 = (E) obj;
}

void test8()
{
    test8();
}

void test9()
{
    try
    {
        catch(StackOverflowError
        {
        }
    }
}

void test10()
{
    int x[] = new int[999999999];
}

void test11()
{
    try
    {
    }
}

```

```

        catch(NoClassDefFoundError
ex)
    {
        System.out.println(ex);
    }
}

// void test12()
// {
//     Class.forName("");
// }

void test13()
{
    try
    {
        Class.forName("");
    }

    catch(ClassNotFoundException ex)
    {
        ex.printStackTrace();
    }
}

// void test14()
// {
//     try
//     {
//         Class.forName("");
        catch(NoClassDefFoundError
//     }
// }

// void test15()
// {
//     try
//     {
//         Class.forName("");
//     }
//     catch(ClassNotFoundException ex)
//     {
//         ex.printStackTrace();
//     }
// }

void test16()
{
    try
    {
        System.out.println("done");
    }
    catch(ClassNotFoundException ex)
    {
        ex.printStackTrace();
    }
}

```

```

//      {
//
//      }
//  }

void test17()
{
    try
    {
        Class.forName("");
        Class.forName("");
        Class.forName("");
        Class.forName("");
    }

    catch(ClassNotFoundException ex)
    {
    }
}

// void test18()
// {
//
//      DriverManager.getConnection("");
//  }

void test18()
{
    try
    {
        DriverManager.getConnection("");
    }
}

// void test19()
// {
//      try
//      {
//      }
//      catch(SQLException ex)
//      {
//      }
//  }

// void test20()
// {
//      try
//      {
//          Class.forName("");
//      }
//      catch(SQLException ex)
//      {
//      }
//  }

```

```

//      }

void test21()
{
    try
    {
        Class.forName("");

DriverManager.getConnection("");
    }
    catch(SQLException ex)
    {

    }

    catch(ClassNotFoundException ex)
    {

    }
}

// void test22()
// {
//     new FileWriter("");
// }

void test23()
{
    try
    {
        new FileWriter("");
    }
}

}

catch(IOException ex)
{

}

// void test24()
// {
//     try
//     {

//     }
//     catch(IOException ex)
//     {

//     }
// }

// void test25()
// {
//     clone();
// }

void test26()
{
    try
    {
        clone();
    }
}

```

			Thread.sleep(1000);
	catch(CloneNotSupportedException	}	
ex)		catch(InterruptedException	
	{		
	}		
	}		
		}	
// void test27			
// {		// void test30()	
// try		// {	
// {		// try	
//		// {	
// }		// }	
//		// }	
	catch(CloneNotSupportedException	// catch(InterruptedException	
ex)		ex)	
// {		// {	
// ex.printStackTrace();		//	
// }		// }	
// }		// }	
// void test28		// void test31()	
// {		// {	
// Thread.sleep(1000);		// DateFormat df =	
// }		//	
		DateFormat.getDateInstance();	
void test29()		// df.parse("");	
{		// }	
try			
{		void test32()	

```

    {
        DateFormat df =
            DateFormat.getDateInstance();
        try
        {
            df.parse("");
        }
        catch(ParseException ex)
        {
            ex.printStackTrace();
        }
    }
// void test33()
// {
//     try
//     {
//     }
//     catch(ParseException ex)
//     {
//     }
// }
// }
// }

```

159. Program

```

package com.lara;
public class F
{

```

```

    public static void main(String[] args)
    {
        System.out.println(1);
        int i = 10/0;
        System.out.println(2);
    }
}

```

160. Program

```

package com.lara;

public class G
{
    public static void main(String[] args)
    {
        System.out.println(1);
        test();
        System.out.println(2);
    }

    static void test()
    {
        System.out.println(3);
        int i = 10/0;
        System.out.println(4);
    }
}

```

161. Program

```

package com.lara;

```

```

public class H
{
    public static void main(String[] args)
    {
        System.out.println(1);
        test();
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        try
        {
            int i = 10/0;
        }
        catch(ArithmeticException
ex)
        {
            System.out.println(4);
        }
        System.out.println(5);
    }
}

```

162. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.println(1);
    }
}

```

```

try
{
    test();
}
catch(ArithmeticException
ex)
{
    System.out.println("exception");
}
System.out.println(2);
}
static void test()
{
    System.out.println(3);
    int i = 10/0;
    System.out.println(4);
}
}

```

163. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try
        {
            test();
        }
    }
}

```



```

        catch(ArithmeticException
ex)
        {

        System.out.println("exception");
        }
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        int i = 10/0;
        System.out.println(4);
    }
}

```

164. Program:

```

package com.lara;

class X
{
    void test1()
    {
        System.out.println(1);
        Y y1 = new Y();
        y1.test2();
        System.out.println(2);
    }
}

class Y

```

```

{
    void test2()
    {
        System.out.println(3);
        try
        {
            Z z1 = new Z();
            z1.test3();
        }
        catch
        (NumberFormatException ex)
        {
            System.out.println("ex");
        }
        System.out.println(4);
    }
}

class Z
{
    void test3()
    {
        System.out.println(5);
        int i =
        Integer.parseInt("abc");
        System.out.println(6);
    }
}

public class J
{
    public static void main(String[] args)

```

```

{
    System.out.println(7);
    X x1 = new X();
    x1.test1();
    System.out.println(8);
}
}

```

165. Program

```

package com.lara;
public class K
{
    public static void main(String[] args)
    {
        System.out.println(1);
        try
        {
            Class.forName("");
        }
        catch(ClassNotFoundException ex)
        {
            System.out.println(ex);
        }
        System.out.println(2);
    }
}

```

166. Program

```

package com.lara;

```

```

public class L
{
    public static void main(String[] args)
    {
        System.out.println(1);
        test();
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        try
        {
            Class.forName("");
        }
        catch(ClassNotFoundException ex)
        {
            System.out.println(ex);
        }
        System.out.println(4);
    }
}

```

167. Program

```

package com.lara;
public class M
{
    public static void main(String[] args)
    {

```

```

        System.out.println(1);
        try
        {
            test();
        }

        catch(ClassNotFoundException ex)
        {

            System.out.println("ex");
        }
        System.out.println(2);
    }
    static void test() throws
    ClassNotFoundException
    {
        System.out.println(3);
        Class.forName("");
        System.out.println(4);
    }
}

```

168. Program

```

package com.lara;
class W
{
    void test() throws
    ClassNotFoundException
    {
        System.out.println(1);
    }
}

```

```

        Class.forName("");
        System.out.println(2);
    }
}
public class N
{
    public static void main(String[] args)
    {
        System.out.println(3);
        W w1 = new W();
        try
        {
            System.out.println(4);
            w1.test();
            System.out.println(5);
        }
        catch(ClassNotFoundException ex)
        {
            System.out.println(6);
        }
        System.out.println(7);
    }
}

```

169. Program

```

package com.lara;
public class O
{
    public static void main(String[] args)
    {
    }
}

```

```
//      try
//      {
//
//      }
//
//      catch(ClassNotFoundException ex)
//      {
//
//
//      }
//
//      }
//
//      void test() throws
//      ClassNotFoundException
//      {
//
//      }
//
//      }
```

170. Program

```
package com.lara;
public class P
{
    static void test() throws
    ClassNotFoundException
    {
    }

    public static void main(String[] args)
    {
        try
        {
```

```
test();
    }

    catch(ClassNotFoundException ex)
    {
    }

    }
}
```

171. Program

```
package com.lara;
import java.io.IOException;
import java.sql.SQLException;
public class Q
{
    static void test1() throws
    SQLException
    {
    }

    static void test2() throws
    IOException
    {
    }

    static void test3() throws
    NullPointerException
    {
```

```

    }

    static void test4()
    {
        try
        {
            test1();
        }
        catch(SQLException ex)
        {

        }
        try
        {
            test2();
        }
        catch(IOException ex)
        {

        }
        test3();
    }
    static void test5()
    {
        try
        {
            test1();
            test2();
        }
        catch(IOException ex)
        {

        }
    }

    static void test6()
        throws SQLException, IOException
    {
        test1();
        test2();
    }
    static void test7() throws Exception
    {
        test1();
        test2();
    }
    void test8() throws Throwable
    {
        test2();
        test1();
    }
    void test9()
    {
        try
        {
            test2();
            test1();
        }
        catch(Exception ex)
    }

```

```

        {
            }
        }
    }
}

```

172. Program

```

package com.lara;
public class R
{
    R() throws ClassNotFoundException
    {
    }
    public static void main(String[] args)
    {
        try
        {
            R r1 = new R();
        }
        catch(ClassNotFoundException ex)
        {
        }
    }
    static void test() throws
    ClassNotFoundException
    {
        R r1 = new R();
    }
}

```

173. Program

```

package com.lara;
import java.sql.SQLException;
public class S
{
    S() throws SQLException
    {
    }
    S(int i) throws SQLException
    {
        this();
    }
    void test1()
    {
        S s1 = null;
        try
        {
            s1 = new S();
            s1 = new S(20);
        }
        catch(SQLException ex)
        {
            ex.printStackTrace();
        }
    }
    void test2() throws SQLException
}

```

```

{
    S s1 = new S();
    s1 = new S(90);
}

void test3() throws Exception
{
    S s1 = new S();
    S s2 = new S(89);
}

```

```

void test4() throws Throwable
{
    S s1 = new S(9);
    s1 = new S();
}

```

```

void test5()
{
    S s1 = null;
    try
    {
        s1 = new S();
        s1 = new S(20);
    }
    catch(Exception ex)
    {
    }

    try
    {

```

```

        s1 = new S(9);
        s1 = new S();
    }
    catch(Throwable t)
    {
    }

}
}

```

174. Program

```

package com.lara;
class V
{
    V() throws
    CloneNotSupportedException
    {
    }
}

public class T extends V
{
    T() throws
    CloneNotSupportedException
    {
    }
}

```

175. Program

```

package com.lara;

```

```

public class AgeIsNegativeException
extends ArithmeticException
{
    public AgeIsNegativeException()
    {

    }
    public
AgeIsNegativeException(String msg)
    {
        super(msg);
    }
}

```

176. Program

```

package com.lara;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.SQLException;

class A
{
    void test1() throws SQLException
    {

    }

    void test2() throws
FileNotFoundException
    {

    }
}

```

```

void test3() throws IOException
{

}

void test4() throws
NullPointerException
{

}

void test5()
{

}
}

```

```

public class B extends A
{
    // void test1()
    // {
    // }

    // void test1() throws SQLException
    // {
    // }

    // void test1() throws Exception
    // {
    // }
}

```



```
// void test1() throws Throwable
// {
//
// }
```

```
// void test1() throws
// ClassNotFoundException
// {
//
// }
```

```
// void test1() throws
// NumberFormatException
// {
//
// }
```

```
// void test2()
// {
//
// }
```

```
// void test2() throws
// FileNotFoundException
// {
//
// }
```

```
// void test2() throws IOException
// {
//
```

```
// }
```

```
// void test2() throws SQLException
// {
//
// }
```

```
// void test2() throws Exception
// {
//
// }
```

```
// void test2() throws
// ArithmeticException
// {
//
// }
```

```
// void test3()
// {
//
// }
```

```
// void test3() throws IOException
// {
//
// }
```

```
// void test3() throws
// FileNotFoundException
// {
```

```
//
//    }

//    void test3() throws
//    ClassNotFoundException
//    {
//
//    }

//    void test3() throws Exception
//    {
//
//    }

//    void test3() throws
//    ClassCastException
//    {
//
//    }

//    void test4()
//    {
//
//    }

//    void test4() throws
//    NullPointerException
//    {
//
//    }
```

```
//    void test4() throws
//    ArithmeticException
//    {
//
//    }

//    void test4() throws
//    ClassNotFoundException
//    {
//
//    }

//    void test5()
//    {
//
//    }

//    void test5() throws
//    NullPointerException
//    {
//
//    }

//    void test5() throws
//    ClassNotFoundException
//    {
//
//    }
```

177. Program

```

package com.lara;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.SQLException;

class A
{
    void test1() throws SQLException
    {

    }

    void test2() throws
FileNotFoundException
    {

    }

    void test3() throws IOException
    {

    }

    void test4() throws
NullPointerException
    {

    }

    void test5()
    {

    }
}

```

```

}

public class B extends A
{
    //    void test1()
    //    {
    //
    //    }

    //    void test1() throws SQLException
    //    {
    //
    //    }

    //    void test1() throws Exception
    //    {
    //
    //    }

    //    void test1() throws Throwable
    //    {
    //
    //    }

    //    void test1() throws
ClassNotFoundException
    //    {
    //
    //    }

    //    void test1() throws
NumberFormatException
}

```

// {	// void test2() throws
//	ArithmeticException
// }	// {
	//
// void test2()	// }
// {	
//	// void test3()
// }	// {
	//
// void test2() throws	// }
FileNotFoundException	
// {	// void test3() throws IOException
//	// {
// }	//
	// }
// void test2() throws IOException	
// {	// void test3() throws
//	FileNotFoundException
// }	// {
	//
// void test2() throws SQLException	// }
// {	
//	// void test3() throws
// }	ClassNotFoundException
	// {
// void test2() throws Exception	//
// {	// }
//	
// }	// void test3() throws Exception
	// {
	//

```
//      }

//      void test3() throws
//      ClassCastException
//      {
//
//      }

//      void test4()
//      {
//
//      }

//      void test4() throws
//      NullPointerException
//      {
//
//      }

//      void test4() throws
//      ArithmeticException
//      {
//
//      }

//      void test4() throws
//      ClassNotFoundException
//      {
//
//      }
```

```
//      void test5()
//      {
//
//      }

//      void test5() throws
//      NullPointerException
//      {
//
//      }

//      void test5() throws
//      ClassNotFoundException
//      {
//
//      }
```

178. Program

```
package com.lara;

public class D
{
    public static void main(String[] args)
    {
        System.out.println(1);
        if(true)
        {
            throw new
            NumberFormatException("some msg");
        }
    }
}
```

```

    }
    System.out.println(2);
}
}

```

179. Program

```

package com.lara;
public class E
{
    public static void main(String[] args)
    {
        System.out.println(1);
        if(true)
        {
            throw new
OutOfMemoryError("some message");
        }
        System.out.println(2);
    }
}

```

180. Program

```

package com.lara;
public class F
{
    public static void main(String[] args)
    throws ClassNotFoundException
    {
        System.out.println(1);
        if(true)
        {

```

```

            throw new
ClassNotFoundException();
        }
        System.out.println(2);
    }
}

```

181. Program

```

package com.lara;
import java.util.Scanner;
public class G
{
    public static void main(String[] args)
    {
        Scanner sc = new
Scanner(System.in);
        System.out.println(1);
        System.out.println("Enter age");
        int age = sc.nextInt();
        if(age <= 0)
        {
            throw new
ArithmeticException();
        }
        System.out.println(2);
        //continue....
    }
}

```

182. Program

```

package com.lara;

```

```

import java.util.Scanner;
public class H
{
    public static void main(String[] args)
    {
        Scanner sc =
            new Scanner(System.in);
        System.out.println(1);
        System.out.println("Enter
age");
        int age = sc.nextInt();
        if(age <= 0)
        {
            throw new
ArithmeticException("Age should be +ve");
        }
        System.out.println(2);
        //continue....
    }
}

```

183. Program

```

package pack1;
public class A
{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert false;
        System.out.println(2);
    }
}

```

```

}

```

184. Program

```

package pack1;
class B
{
    public static void main(String[] args)
    {
        System.out.println(1);
        int i = 0;
        assert i > 0;
        System.out.println(2);
    }
}

```

185. Program

```

package pack1;
class C
{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert true;
        System.out.println(2);
    }
}

```

186. Program

```

package pack1;
class E
{
    public static void main(String[] args)

```

```

    {
        System.out.println(1);
        assert false : "error occurred";
        System.out.println(2);
    }
}

```

187. Program

```

package pack1;
class F
{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert false : 1000;
        System.out.println(2);
    }
}

```

188. Program

```

package pack1;
class G
{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert false : false;
        System.out.println(2);
    }
}

```

189. Program

```

package pack1;
class H

```

```

{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert false : test();
        System.out.println(2);
    }
    static int test()
    {
        return 100;
    }
}

```

190. Program

```

package pack1;
class I
{
    public static void main(String[] args)
    {
        System.out.println(1);
        assert test();
        System.out.println(2);
    }
    static boolean test()
    {
        return false;
    }
}

```

191. Program

```

package pack1;
class J
{
    public static void main(String[] args)

```



```

    {
        System.out.println("Hello
        World!");
        int assert = 10;
        System.out.println(assert);
    }
}

```

192. Program

```

package pack1;
class D
{
    void test1()
    {
        System.out.println("test1-
        begin");
        assert false;
        System.out.println("test1-
        end");
    }
}
class Manager
{
    public static void main(String[] args)
    {
        System.out.println("main
        begin");
        assert false;
        D d1 = new D();
        d1.test1();
        System.out.println("main
        end");
    }
}

```

```

    }
}

193. Program

package com.lara;
import java.util.Scanner;
public class I
{
    public static void main(String[] args)
    {
        Scanner sc =
        new Scanner(System.in);
        System.out.println(1);
        System.out.println("Enter
        age");
        int age = sc.nextInt();
        if(age <= 0)
        {
            throw new
            AgeIsNegativeException("age should not e -
            ve");
        }
        System.out.println(2);
        //continue....
    }
}

```

194. Program

```

package com.lara;
public class Manager
{
    public static void main(String[] args)
    {

```

```

    {
        System.out.println(1);
        test();
        System.out.println(2);
    }
    static void test()
    {
        System.out.println(3);
        System.out.println(4);
    }
}

```

195. Program

```

package com.lara;
public class Manager1
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }

        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

196. Program

```

package com.lara;
class Thread1 extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager2
{
    public static void main(String[] args)
    {
        Thread1 t1 = new Thread1();
        t1.start();
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

197. Program

```

package com.lara;
class ThreadA extends Thread
{
    public void run()

```

```

        {
            for(int i = 0; i < 1000; i++)
            {
                System.out.println(i);
            }
        }
    }
}

class ThreadB extends Thread
{
    public void run()
    {
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager3
{
    public static void main(String[] args)
    {
        ThreadA t1 = new
ThreadA();

        t1.start();

        ThreadB t2 = new ThreadB();
        t2.start();
        //t2.setDaemon(on)
    }
}

```

```

        for(int i = 2000; i < 3000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

198. Program

```

package com.lara;
class B extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager4
{
    public static void main(String[] args)
    {
        B b1 = new B();
        b1.start();
        System.out.println("done");
    }
}

```

199. Program

```
package com.lara;

class C implements Runnable
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager5
{
    public static void main(String[] args)
    {
        C c1 = new C();
        Thread t1 = new Thread(c1);
        t1.start();

        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}
```

200.Program

```
package com.lara;
```

```
class D extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager6
{
    public static void main(String[] args)
    {
        D d1 = new D();
        d1.start();

        D d2 = new D();
        d2.start();

        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}
```

201.Program

```
package com.lara;
```

```

class D extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}

public class Manager6
{
    public static void main(String[] args)
    {
        D d1 = new D();
        d1.start();
        D d2 = new D();
        d2.start();
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

202. Program

```

package com.lara;

class E implements Runnable
{

```

```

    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }

    public class Manager7
    {
        public static void main(String[] args)
        {
            E e1 = new E();
            Thread t1 = new Thread(e1);
            t1.start();
            Thread t2 = new Thread(e1);
            t2.start();
            for(int i = 1000; i < 2000;
i++)
            {
                System.out.println(i);
            }
        }
    }
}

```

203. Program

```

package com.lara;

class F extends Thread
{
    public void run()

```

```

        {
            for(int i = 0; i < 1000; i++)
            {
                System.out.println(i);
                start();
            }
        }
    }

    public class Manager8
    {
        public static void main(String[] args)
        {
            F f1 = new F();
            f1.start();
            for(int i = 1000; i < 2000;
i++)
            {
                System.out.println(i);
            }
        }
    }

```

204. Program

```

package com.lara;
class G extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {

```

```

            System.out.println(i);
        }
    }

    void startThread()
    {
        start();
    }
}

    public class Manager9
    {
        public static void main(String[] args)
        {
            G g1 = new G();
            g1.startThread();
            for(int i = 1000; i < 2000;
i++)
            {
                System.out.println(i);
            }
        }
    }

```

205. Program

```

package com.lara;
class H extends Thread
{
    H()
    {
        start();
    }
}

```

```

        public void run()
        {
            for(int i = 0; i < 1000; i++)
            {
                System.out.println(i);
            }
        }
    }

    public class Manager10
    {
        public static void main(String[] args)
        {
            H h1 = new H();
            //h1.start();
            for(int i = 1000; i < 2000; i
            ++))
            {
                System.out.println(i);
            }
        }
    }
}

```

206. Program

```

package com.lara;
class I extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)

```

```

        {
            System.out.println(i);
        }
    }

    public class Manager11
    {
        public static void main(String[] args)
        {
            I obj = new I();
            obj.run();
            obj.start();
            for(int i = 1000; i < 2000;
            i++)
            {
                System.out.println(i);
            }
        }
    }
}

```

207. Program

```

package com.lara;
public class Manager12
{
    static class A extends Thread
    {
        public void run()
        {
            for(int i = 0; i < 1000; i ++))
            {

```

```

        System.out.println(i);
    }
}
static class B implements Runnable
{
    public void run()
    {
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
    public static void main(String[] args)
    {
        A a1 = new A();
        a1.start();
        B b1 = new B();
        Thread t1 = new Thread(b1);
        t1.start();
        for(int i = 2000; i < 3000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

208. Program

```

package com.lara;
public class Manager13
{
    public static void main(String[] args)
    {
        class A extends Thread
        {
            public void run()
            {
                for(int i = 0; i < 1000; i++)
                {
                    System.out.println(i);
                }
            }
        }
        class B implements Runnable
        {
            public void run()
            {
                for(int i = 1000; i < 2000;
i++)
                {
                    System.out.println(i);
                }
            }
        }
        A a1 = new A();
        a1.start();
        B b1 = new B();
    }
}

```



```

        Thread t1 = new Thread(b1);
        t1.start();
        for(int i = 2000; i < 3000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

209. Program

```

package com.lara;
public class Manager14
{
    public static void main(String[] args)
    {
        Thread t1 = new Thread()
        {
            public void run()
            {
                for(int i = 0; i < 1000; i++)
                {
                    System.out.println(i);
                }
            }
        };
        t1.start();

        Runnable r1 = new
Runnable()
    {

```

```

        public void run()
        {
            for(int i = 1000; i < 2000;
i++)
            {
                System.out.println(i);
            }
        }
    };
    Thread t2 = new Thread(r1);
    t2.start();
    for(int i = 2000; i < 3000;
i++)
    {
        System.out.println(i);
    }
}

```

210. Program

```

package com.lara;
public class Manager15
{
    public static void main(String[] args)
    {
        new Thread()
        {
            public void run()
            {
                for(int i = 0; i < 1000; i++)

```

```

        {
            System.out.println(i);
        }
    }
}.start();
new Thread(new Runnable()
{
    public void run()
    {
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}).start();

for(int i = 2000; i < 3000;
i++)
{
    System.out.println(i);
}
}

```

211. Program

```

package com.lara;
class A extends Thread
{
    public void run()

```

```

{
    for(int i = 0; i < 10; i++)
    {
        System.out.println(i);
        try
        {
            sleep(1000);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
    }
}

public class Manager1
{
    public static void main(String[] args)
    {
        A a1 = new A();
        a1.start();
        System.out.println("done");
    }
}

```

212. Program

```

package com.lara;
class B implements Runnable
{

```

```

public void run()
{
    for(int i = 0; i < 10; i++)
    {
        System.out.println(i);
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);
        }
    }
}

public class Manager2
{
    public static void main(String[] args)
    {
        B b1 = new B();
        Thread t1 = new Thread(b1);
        t1.start();
        for(int i = 10; i < 20; i++)
        {
            System.out.println(i);
            try
            {
                Thread.sleep(1000);

```

```

        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);
        }
    }
}

213. Program

package com.lara;
class Util
{
    static void sleep(long millis)
    {
        try
        {
            Thread.sleep(millis);
        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);
        }
    }
}

class C extends Thread

```

```

{
    public void run()
    {
        for(int i = 0; i < 20; i++)
        {
            System.out.println(i);
            Util.sleep(1000);
        }
    }
}
class D implements Runnable
{
    public void run()
    {
        for(int i = 20; i < 40; i++ )
        {
            System.out.println(i);
            Util.sleep(1000);
        }
    }
}
public class Manager3
{
    public static void main(String[] args)
    {
        C c1 = new C();
        c1.start();
        D d1 = new D();
        Thread t1 = new Thread(d1);
        t1.start();
        for(int i = 40; i < 60; i++)

```

```

{
    System.out.println(i);
    Util.sleep(1000);
}
}
}

```

214. Program

```

package com.lara;
public class Manager4
{
    public static void main(String[] args)
    {
        Thread t1 = Thread.currentThread();
        t1.setName("initiator");
        System.out.println(t1.getId());
        System.out.println(t1.getName());
        System.out.println(t1.getPriority());
        System.out.println(t1.isDaemon());
    }
}

```

215. Program

```

package com.lara;
class E extends Thread
{
}
public class Manager5

```

```

{
    public static void main(String[] args)
    {
        E e1 = new E();
        e1.setName("first thread");
        e1.start();
        E e2 = new E();
        e2.setName("second thread");
        e2.start();
        System.out.println(e1.getId());
        System.out.println(e1.getName());
        System.out.println(e1.getPriority());
        System.out.println(e1.isDaemon());
        System.out.println("----");
        System.out.println(e2.getId());
        System.out.println(e2.getName());
        System.out.println(e2.getPriority());
        System.out.println(e2.isDaemon());
    }
}

```

216. Program

```

package com.lara;
class F extends Thread
{
    F(String s1)
    {
        super(s1);
    }
}

```

```

class G implements Runnable
{
    public void run()
    {
    }
}

public class Manager6
{
    public static void main(String[] args)
    {
        F f1 = new F("first");
        f1.start();
        G g1 = new G();
        Thread t1 = new Thread(g1,
            "second");
        t1.start();
        System.out.println(f1.getName());
        System.out.println(t1.getName());
    }
}

```

217. Program

```

package com.lara;
public class Manager7
{
    public static void main(String[] args)
    {
        Thread t1 = Thread.currentThread();
        System.out.println(t1.getPriority());
        t1.setPriority(10);
        System.out.println(t1.getPriority());
    }
}

```

```
    }
}
```

218. Program

```
package com.lara;
public class Manager8
{
    public static void main(String[] args)
    {
        class A extends Thread
        {
            }
        A a1 = new A();
        System.out.println(a1.getPriority());
        Thread t1 = Thread.currentThread();
        t1.setPriority(Thread.MAX_PRIORITY);
        A a2 = new A();
        System.out.println(a2.getPriority());
    }
}
```

219. Program

```
package com.lara;
class H extends Thread
{
    public void run()
    {
        for(int i = 0; i < 3000; i++)
```

```
    {
        System.out.println(i);
    }
}

public class Manager9
{
    public static void main(String[] args)
    {
        H obj = new H();
        //obj.setDaemon(true);
        obj.start();
        System.out.println("done");
    }
}
```

220. Program

```
package com.lara;
class I extends Thread
{
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
        System.out.println("child
end");
    }
}
```

```

public class Manager10
{
    public static void main(String[] args)
    {
        I obj = new I();
        obj.start();
        try
        {
            obj.join();
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

221. Program

```

package com.lara;
import java.util.Scanner;
class J extends Thread
{
    public void run()
    {
        int counter = 0;

```

```

        while(! isInterrupted())
        {
            counter ++;
            System.out.println(counter);
            try
            {
                sleep(1000);
            }
            catch(InterruptedException
ex)
            {
                System.out.println(ex);
                break;
            }
        }
    }
}

public class Manager11
{
    public static void main(String[] args)
    {
        J obj = new J();
        obj.start();
        Scanner sc = new
Scanner(System.in);
        String decider;
        do
        {
            try
            {

```

```

        Thread.sleep(20000);
    }
    catch(InterruptedException
ex)
    {
        ex.printStackTrace();
    }
    System.out.println("continue?(y/n)");
    decider = sc.next();
    }while("y".equals(decider));
    obj.interrupt();
    }
}

```

222. Program

```

package com.lara.pack1;
class A
{
    synchronized void test1()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
    synchronized void test2()
    {
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

```

    }
}
}
class B extends Thread
{
    A a1;
    B(A a1)
    {
        this.a1 = a1;
    }
    public void run()
    {
        a1.test1();
    }
}
class C extends Thread
{
    A a1;
    C(A a1)
    {
        this.a1 = a1;
    }
    public void run()
    {
        a1.test2();
    }
}
public class Manager1
{
    public static void main(String[] args)
    {
    }
}

```



```

        A a1 = new A();
        B b1 = new B(a1);
        C c1 = new C(a1);
        b1.start();
        c1.start();
    }
}

```

223. Program

```

package com.lara.pack2;

class Shared
{
    void test1()
    {
        Thread t1 = Thread.currentThread();
        for(int i = 0; i < 1000; i++)
        {
            System.out.println("test1:" +
t1.getName() + ":" + i);
        }
    }

    void test2()
    {
        Thread t1 = Thread.currentThread();
        for(int i = 0; i < 1000; i++)
        {
            System.out.println("test2:" +
t1.getName() + ":" + i);
        }
    }
}

```

```

        synchronized void test3()
        {
            Thread t1 = Thread.currentThread();
            for(int i = 0; i < 1000; i++)
            {
                System.out.println("test3:" +
t1.getName() + ":" + i);
            }
        }

        synchronized void test4()
        {
            Thread t1 = Thread.currentThread();
            for(int i = 0; i < 1000; i++)
            {
                System.out.println("test4:" +
t1.getName() + ":" + i);
            }
        }

        static void test5()
        {
            Thread t1 = Thread.currentThread();
            for(int i = 0; i < 1000; i++)
            {
                System.out.println("test5:" +
t1.getName() + ":" + i);
            }
        }

        static void test6()
        {
            Thread t1 = Thread.currentThread();
            for(int i = 0; i < 1000; i++)

```

```

        {
            System.out.println("test6:" +
t1.getName() + ":" + i);
        }
    }

    synchronized static void test7()
    {
        Thread t1 = Thread.currentThread();
        for(int i = 0; i < 1000; i++)
        {

            System.out.println("test7:" +
t1.getName() + ":" + i);
        }
    }

    synchronized static void test8()
    {
        Thread t1 = Thread.currentThread();
        for(int i = 0; i < 1000; i++)
        {
            System.out.println("test8:" +
t1.getName() + ":" + i);
        }
    }
}

class Thread1 extends Thread
{
    Shared s1;
    Thread1(Shared s1)
    {
        this.s1 = s1;
    }
}

```

```

    }

    public void run()
    {
        s1.test3(); //(A)
    }
}

class Thread2 extends Thread
{
    Shared s1;
    Thread2(Shared s1)
    {
        this.s1 = s1;
    }
    public void run()
    {
        s1.test3(); //(B)
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        Shared s1 = new Shared();
        Shared s2 = new Shared();
        Thread1 t1 = new Thread1(s1); //(C)
        Thread2 t2 = new Thread2(s1); //(D)
        t1.start();
        t2.start();
    }
}

```

Case 1:

A-test1

B-test1

C-s1

D-s1

Case 2:

A-test2

B-test2

C-s1

D-s1

Case 3:

A-test1

B-test2

C-s1

D-s1

Case 4:

A-test2

B-test3

C-s1

D-s1

Case 5:

A-test3

B-test3

C-s1

D-s1

Case 6:

A-test3

B-test4

C-s1

D-s1

Case 7:

A-test4

B-test4

C-s1

D-s1

Case 8:

A-test4

B-test5

C-s1

D-s1

Case 9:

A-test5

B-test5

C-s1

D-s1

Case 10:

A-test5

B-test6

C-s1

D-s1

Case 11:

A-test5

B-test6

C-s1

D-s2

Case 12:

A-test6

B-test7

C-s1

D-s2

Case 13:

A-test7

B-test7

C-s1

D-s2

Case 14:

A-test7

B-test8

C-s1

D-s2

Case 15:

A-test4

B-test8

C-s1

D-s2

Case 16:

A-test1

B-test5

C-s1

D-s2

Case 17:

A-test8

B-test8

C-s1

D-s1

Case18 :

A-test8

B-test8

C-s1

D-s2

224. Program

```

package com.lara.pack3;
import
java.lang.management.ManagementFactory;
import java.lang.management.ThreadInfo;
import
java.lang.management.ThreadMXBean;
class Shared
{
    synchronized void test1(Shared s)
    {
        System.out.println("test1 begin");
        Util.sleep(1000);
        s.test2(this);
        System.out.println("test1 end");
    }
    synchronized void test2(Shared s)
    {
        System.out.println("test2 begin");
        Util.sleep(1000);
        s.test1(this);
        System.out.println("test2 end");
    }
}

```

```

    }
}
class A extends Thread
{
    Shared s1, s2;
    A(Shared s1, Shared s2)
    {
        this.s1 = s1;
        this.s2 = s2;
    }
    public void run()
    {
        s1.test1(s2);
    }
}
class B extends Thread
{
    Shared s1, s2;
    B(Shared s1, Shared s2)
    {
        this.s1 = s1;
        this.s2 = s2;
    }
    public void run()
    {
        s2.test2(s1);
    }
}
class Util
{
    static void sleep(long millis)

```

```

        {
            try
            {
                Thread.sleep(millis);
            }
            catch (InterruptedException
ex)
            {
                ex.printStackTrace();
            }
        }
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        Shared s1 = new Shared();
        Shared s2 = new Shared();
        A a1 = new A(s1, s2);
        a1.start();
        B b1 = new B(s1, s2);
        b1.start();
        Util.sleep(2000);
        ThreadMXBean tx =
ManagementFactory.getThreadMXBean();
        long ids[] =
tx.findDeadlockedThreads();
        if (ids != null)
        {
            System.out.println("dead
locked threads are");

```

```

        ThreadInfo ti[] =
tx.getThreadInfo(ids);
        ThreadInfo thInfo = null;
        for (int i = 0; i < ti.length;
i++)
        {
            thInfo = ti[i];
            System.out.println(thInfo.getThread
Name());
        }
        else
        {
            System.out.println("no threads are
under locked");
        }
    }
}

```

225. Program

```

package com.lara.pack4;
class A
{
    synchronized void test1()
    {
        System.out.println("test1
begin");
        try
        {
            wait();

```

```

        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        System.out.println("test1
end");
    }
    synchronized void test2()
    {
        notifyAll();
    }
}
class Thread1 extends Thread
{
    A a1;
    Thread1(A a1)
    {
        this.a1 = a1;
    }
    public void run()
    {
        a1.test1();
    }
}
class Thread2 extends Thread
{
    A a1;
    Thread2(A a1)
    {
        this.a1 = a1;
    }
    public void run()
    {
        a1.test1();
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        A a1 = new A();
        Thread1 t1 = new
        Thread1(a1);
        Thread2 t2 = new
        Thread2(a1);
        t1.start();
        t2.start();
        try
        {
            Thread.sleep(20000);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        System.out.println("about to
release");
        new A().test2();
    }
}

```

}

226. Program

package com.lara.pack5;

class A

{

synchronized void test1()

{

try

{

wait();

}

catch (InterruptedException

ex)

{

ex.printStackTrace();

}

}

synchronized void test2()

{

//notify();

notifyAll();

}

}

class B extends Thread

{

A obj;

B(A obj)

{

this.obj = obj;

}

public void run()

{

System.out.println("B-run begin");

obj.test1();

System.out.println("B-run end");

}

}

class C extends Thread

{

A obj;

C(A obj)

{

this.obj = obj;

}

public void run()

{

System.out.println("C-run begin");

obj.test1();

System.out.println("C-run end");

}

}

class D extends Thread

{

A obj;

D(A obj)

{

this.obj = obj;

}

public void run()

{

System.out.println("D-run begin");


```

        obj.test1();
        System.out.println("D-run end");
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        A a1 = new A();
        A a2 = new A();
        B b1 = new B(a1);
        C c1 = new C(a1);
        D d1 = new D(a1);
        b1.start();
        c1.start();
        d1.start();

        try
        {
            Thread.sleep(30000);
        }
        catch (InterruptedException
ex)
        {
            ex.printStackTrace();
        }

        System.out.println("about to
release");

        //a1.test2();
        a2.test2();
    }
}

```

227. Program

```

    }
}

package com.lara.pack6;
class A
{
    void test1()
    {
        //some stmts
        synchronized (this) //mutex
        {
            try
            {
                wait();
            }
            catch (InterruptedException
ex)
            {
                ex.printStackTrace();
            }
        }
        //some more stmts
    }

    void test2()
    {
        //some stmts
        synchronized (this)
        {
            notifyAll();
        }
    }
}

```

```

        //some other stmts
    }
}
class B extends Thread
{
    A obj;
    B(A obj)
    {
        this.obj = obj;
    }
    public void run()
    {
        System.out.println("run begin");
        obj.test1();
        System.out.println("run end");
    }
}
public class Manager
{
    public static void main(String[] args)
    {
        A a = new A();
        B b = new B(a);
        b.start();
        try
        {
            Thread.sleep(10000);
        }
        catch (InterruptedException
ex)
    {

```

```

        ex.printStackTrace();
    }
    a.test2();
}
}

```

228. Program

```

package com.lara.pack7;
class Thread1 extends Thread
{
    public synchronized void run()
    {
        System.out.println("begin");
        try
        {
            wait();
        }
        catch (InterruptedException
ex)
    {
        ex.printStackTrace();
    }
    System.out.println("end");
}
}
public class Manager
{
    public static void main(String[] args)
    {
        Thread1 t1 = new Thread1();
        t1.start();

```

```

        try
        {
            Thread.sleep(20000);
        }
        catch (InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        synchronized (t1)
        {
            t1.notify();
        }
    }
}

```

229. Program

```

package com.lara.pack8;
public class Manager1
{
    public static void main(String[] args)
    {
        Thread.State states[] =
            Thread.State.values();
        for (Thread.State state : states)
        {
            System.out.println(state);
        }
    }
}

```

230. Program

```

package com.lara.pack8;
class A extends Thread
{
    public void run()
    {
        for (int i = 0; i < 3000; i++)
        {
            System.out.println(i);
        }
    }
}
public class Manager2
{
    public static void main(String[] args)
    {
        A a1 = new A();
        System.out.println("a:" +
a1.getState());
        a1.start();
        System.out.println("b:" +
a1.getState());
        try
        {
            Thread.sleep(1);
        }
        catch (InterruptedException
ex)
    }
}

```

```

        {
            System.out.println(ex);
        }
        System.out.println("c:" +
a1.getState());
        try
        {
            Thread.sleep(10000);
        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);
        }
        System.out.println("d:" +
a1.getState());
    }
}

```

231. Program

```

package com.lara.pack8;
class B extends Thread
{
    public void run()
    {
        System.out.println("begin");
        try
        {

```

```

            Thread.sleep(10000);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        System.out.println("end");
    }
}
public class Manager3
{
    public static void main(String[] args)
    {
        B b1 = new B();
        b1.start();
        try
        {
            Thread.sleep(5000);
        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);
        }
        System.out.println(b1.getState());
    }
}

```

232. Program

```

package com.lara.pack8;
class C extends Thread
{
    public synchronized void run()
    {
        System.out.println("begin");
        try
        {
            wait();
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        System.out.println("end");
    }
}

public class Manager4
{
    public static void main(String[] args)
    {
        C c1 = new C();
        c1.start();
        try
        {
            Thread.sleep(5000);
        }
        catch(InterruptedException
ex)
        {

```

233. Program

```

        ex.printStackTrace();
    }

    System.out.println(c1.getState());
    synchronized (c1)
    {
        c1.notify();
    }
}

package com.lara.pack8;
class D extends Thread
{
    Thread main;
    D(Thread main)
    {
        this.main = main;
    }
    public void run()
    {
        try
        {
            Thread.sleep(2000);
        }
        catch(InterruptedException
ex)
        {
            System.out.println(ex);

```

```

    }

    System.out.println(main.getState());
}

public class Manager5
{
    public static void main(String[] args)
    {
        Thread main =
Thread.currentThread();
        D d1 = new D(main);
        d1.start();
        try
        {
            d1.join();
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        System.out.println("end");
    }
}

```

234. Program

```

package com.lara.pack8;
class Shared
{
    synchronized void test1(Shared s1)

```

```

{
    System.out.println(1);
    Util.sleep(1000);
    s1.test2(this);
    System.out.println(2);
}

synchronized void test2(Shared s1)
{
    System.out.println(3);
    Util.sleep(1000);
    s1.test1(this);
    System.out.println(4);
}

class Util
{
    static void sleep(long millis)
    {
        try
        {
            Thread.sleep(millis);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
    }
}

class E extends Thread
{

```

```

        Shared s1, s2;
        E(Shared s1, Shared s2)
        {
            this.s1 = s1;
            this.s2 = s2;
        }
        public void run()
        {
            s1.test1(s2);
        }
    }
class F extends Thread
{
    Shared s1, s2;
    F(Shared s1, Shared s2)
    {
        this.s1 = s1;
        this.s2 = s2;
    }
    public void run()
    {
        s2.test2(s1);
    }
}
public class Manager6
{
    public static void main(String[] args)
    {
        Shared s1 = new Shared();
        Shared s2 = new Shared();
        E e1 = new E(s1, s2);

```

```

        F f1 = new F(s1, s2);
        e1.start();
        f1.start();
        Util.sleep(10000);
        System.out.println(e1.getState());
        System.out.println(f1.getState());
    }
}

```

235. Program

```

package com.lara.pack8;
class G extends Thread
{
    public void run()
    {
        System.out.println("begin");
        try
        {
            Thread.sleep(10000);
        }
        catch (InterruptedException
            ex)
        {
            ex.printStackTrace();
        }
        System.out.println("end");
    }
}
public class Manager7
{
    public static void main(String[] args)

```

```

    {
        G g1 = new G();
        g1.start();
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }
        g1.stop();
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException
ex)
        {
            ex.printStackTrace();
        }

        System.out.println(g1.getState());
    }

```

236. Program

```

package com.lara.pack9;
class Util

```

```

    {
        static void sleep(long millis)
        {
            try
            {
                Thread.sleep(millis);
            }
            catch(InterruptedException
ex)
            {
                ex.printStackTrace();
            }
        }
    }

    class Test
    {
        int i;
    }

    class A extends Thread
    {
        Test t;
        A(Test t)
        {
            this.t = t;
        }

        public void run()
        {
            System.out.println("1:" + t.i);
            t.i = 10;
            Util.sleep(500);
            System.out.println("2:" + t.i);
        }
    }

```



```

        t.i = 20;
        Util.sleep(500);
        System.out.println("3:" + t.i);
        t.i = 30;
        Util.sleep(500);
        System.out.println("4:" + t.i);
        t.i = 40;
    }
}
class B extends Thread
{
    Test t;
    B(Test t)
    {
        this.t = t;
    }
    public void run()
    {
        System.out.println("5:" + t.i);
        t.i = 50;
        Util.sleep(500);
        System.out.println("6:" + t.i);
        t.i = 60;
        Util.sleep(500);
        System.out.println("7:" + t.i);
        t.i = 70;
        Util.sleep(500);
        System.out.println("8:" + t.i);
        t.i = 80;
    }
}

```

```

public class Manager
{
    public static void main(String[] args)
    {
        Test t1 = new Test();
        t1.i = 90;
        A a1 = new A(t1);
        a1.start();
        Util.sleep(250);
        B b1 = new B(t1);
        b1.start();
        Util.sleep(40000);
        System.out.println("9:" +
            t1.i);
    }
}

```

237. Program

```

package com.lara.pack10;
class Util
{
    static void sleep(long millis)
    {
        try
        {
            Thread.sleep(millis);
        }
        catch (InterruptedException
            ex)
        {

```

```

        ex.printStackTrace();
    }
}

class A extends Thread
{
    ThreadLocal t;
    A(ThreadLocal t)
    {
        this.t = t;
    }
    public void run()
    {
        System.out.println("1:" + t.get());
        t.set(10);
        Util.sleep(500);
        System.out.println("2:" + t.get());
        t.set(20);
        Util.sleep(500);
        System.out.println("3:" + t.get());
        t.set(30);
        Util.sleep(500);
        System.out.println("4:" + t.get());
        t.set(40);
    }
}

class B extends Thread
{
    ThreadLocal t;
    B(ThreadLocal t)
    {

```

```

        this.t = t;
    }
    public void run()
    {
        System.out.println("5:" + t.get());
        t.set(50);
        Util.sleep(500);
        System.out.println("6:" + t.get());
        t.set(60);
        Util.sleep(500);
        System.out.println("7:" + t.get());
        t.set(70);
        Util.sleep(500);
        System.out.println("8:" + t.get());
        t.set(80);
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        ThreadLocal t1 = new
ThreadLocal();
        t1.set(90);
        A a1 = new A(t1);
        a1.start();
        Util.sleep(250);
        B b1 = new B(t1);
        b1.start();
        Util.sleep(40000);
        System.out.println("9:" + t1.get());
    }
}

```

```

    }
}

```

238. Program

```

package com.lara.pack12;
class A extends Thread
{
    A(ThreadGroup tg, String name)
    {
        super(tg, name);
    }
    public void run()
    {
        for(int i = 0; i < 1000; i++)
        {
            System.out.println(i);
        }
    }
}
class B implements Runnable
{
    public void run()
    {
        for(int i = 1000; i < 2000;
i++)
        {
            System.out.println(i);
        }
    }
}

```

```

public class Manager
{
    public static void main(String[] args)
    {
        ThreadGroup tg =
            new
ThreadGroup("first group");
        A a1 = new A(tg, "first
thread");
        A a2 = new A(tg, "2nd
thread");
        B b1 = new B();
        Thread t1 = new Thread(tg,
b1, "3rd thread");
        Thread t2 = new Thread(tg,
b1, "4th thread");
        a1.start();
        a2.start();
        //t1.start();
        t2.start();
        tg.stop();
    }
}

```

239. Program

```

package com.lara.pack13;
class A
{
    //some members
}
class B extends Thread

```

```

{
    //several attributues
    public void run()
    {
        //some stmts
    }
}
class C extends A implements Runnable
{
    //several attributes
    public void run()
    {
        //some stmts
    }
}
public class Manager
{
    public static void main(String[] args)
    {
        B b1 = new B();
        b1.start();
        B b2 = new B();
        b2.start();
        C c1 = new C();
        Thread t1 = new Thread(c1);
        Thread t2 = new Thread(c1);
        t1.start();
        t2.start();
    }
}

```

240. Program

```

package com.lara;
import java.util.ArrayList;
class Util
{
    static void sleep(long millis)
    {
        try
        {
            Thread.sleep(millis);
        }
        catch (InterruptedException
ex)
        {
            ex.printStackTrace();
        }
    }
}
class ModelThread extends Thread
{
    public synchronized void
goToWait()
    {
        try
        {
            wait();
        }
        catch (InterruptedException
ex)
        {
            ex.printStackTrace();
        }
    }
}

```

```

    }
    public synchronized void release()
    {
        notify();
    }
    public void run()
    {
        while(true)
        {
            goAndWait();
            //main task starting
            for(int i = 0; i < 10;
i++)
            {
                System.out.println(getName() + ":" +
i);
                Util.sleep(1000);
            }
            //main task end
            release();
        }
    }
}
class ThreadPoolManager
{
    private ArrayList pool = new
ArrayList();
    public void init()
    {
        ModelThread th = null;
        for(int i = 0; i < 10; i++)
    {
        th = new
        ModelThread();
        th.start();
        pool.add(th);
    }
}

    public ModelThread getThread()
    {
        ModelThread th = null;
        if(pool.size() > 0)
        {
            th = (ModelThread)
pool.remove(0);
        }
        else
        {
            th = new
            ModelThread();
            th.start();
        }
        return th;
    }

    public void setThread(ModelThread
th)
    {
        if(pool.size() < 10)
        {
            pool.add(th);
        }
        else

```

```

        {
            th.stop();
            th = null;
        }
    }

    public void release()
    {
        ModelThread th = null;
        for(int i = 0; i < pool.size(); )
        {
            th =
(ModelThread)pool.remove(0);
            th.stop();
        }
    }
}

class User1 extends Thread
{
    ThreadPoolManager pm = null;
    User1(ThreadPoolManager pm)
    {
        this.pm = pm;
    }

    public void run()
    {
        while(true)
        {
            ModelThread th =
pm.getThread();

            th.release();
            th.goToWait();
        }
    }
}

        pm.setThread(th);
        Util.sleep(1000);
    }
}

class User2 extends Thread
{
    ThreadPoolManager pm = null;
    User2(ThreadPoolManager pm)
    {
        this.pm = pm;
    }

    public void run()
    {
        while(true)
        {
            ModelThread th =
pm.getThread();

            th.release();
            th.goToWait();
            pm.setThread(th);
            Util.sleep(1000);
        }
    }
}

class User3 extends Thread
{
    ThreadPoolManager pm = null;
    User3(ThreadPoolManager pm)
    {
        this.pm = pm;
    }
}

```

```

    }
    public void run()
    {
        while(true)
        {
            ModelThread th =
pm.getThread();

            th.release();
            th.goToWait();
            pm.setThread(th);
            Util.sleep(1000);
        }
    }
}

public class Manager
{
    public static void main(String[] args)
    {
        ThreadPoolManager pm =
new ThreadPoolManager();
        pm.init();
        User1 u1 = new User1(pm);
        User2 u2 = new User2(pm);
        User3 u3 = new User3(pm);
        u1.start();
        u2.start();
        u3.start();
        Util.sleep(500000);
        u1.stop();
        u2.stop();
        u3.stop();
    }
}

```

```

        pm.release();
        System.out.println("End of
the Game");
    }
}

```

241. Program

```

package com.lara;
class A
{
    int i;
}

public class Manager1
{
    public static void main(String[] args)
    {
        A a1 = new A();
        a1.i = 20;
        System.out.println(a1);
    }
}

```

242. Program

```

package com.lara;
class B
{
    int i;
    B(int i)
    {

```

```

        this.i = i;
    }
}
public class Manager2
{
    public static void main(String[] args)
    {
        B b1 = new B(10);
        String s1 = b1.toString();
        System.out.println(s1);
        System.out.println(b1.toString());
        System.out.println(b1);
    }
}

```

243. Program

```

package com.lara;
class C
{
    int i;
    C(int i)
    {
        this.i = i;
    }
    public String toString()
    {
        return "i = " + i;
    }
}
public class Manager3

```

```

{
    public static void main(String[] args)
    {
        C c1 = new C(90);
        System.out.println(c1);
        C c2 = new C(20);
        System.out.println(c2);
    }
}

```

244. Program

```

package com.lara;
class D
{
    int i;
    D(int i)
    {
        this.i = i;
    }
}
public class Manager4
{
    public static void main(String[] args)
    {
        D d1 = new D(90);
        D d2 = new D(90);
        D d3 = d2;
        D d4 = d1;
        System.out.println(d1);
        System.out.println(d2);
    }
}

```



```

        System.out.println(d3);
        System.out.println(d4);
    }
}

```

245. Program

```

package com.lara;
class E
{
    int i, j;
    E(int i, int j)
    {
        this.i = i;
        this.j = j;
    }
    public String toString()
    {
        return "i = " + i + " & j = " +
j;
    }
}
public class Manager5
{
    public static void main(String[] args)
    {
        E e1 = new E(1, 2);
        E e2 = new E(11, 25);
        System.out.println(e1);
        System.out.println(e2);
    }
}

```

```

    }
}

```

246. Program

```

package com.lara;
class F
{
    String s1;
    int i;
    F(String s1, int i)
    {
        this.s1 = s1;
        this.i = i;
    }
    public String toString()
    {
        return "s1 = " + s1 + ", i = " +
i;
    }
}
public class Manager6
{
    public static void main(String[] args)
    {
        F f1 = new F("abc", 22);
        F f2 = new F("abc", 22);
        System.out.println(f1);
        System.out.println(f2);
    }
}

```

247. Program

```

package com.lara;
class G
{
    int i;
    G(int i)
    {
        this.i = i;
    }
    public String toString()
    {
        return "i = " + i;
    }
}
class H
{
    G g1;
    int j;
    H(G g1, int j)
    {
        this.g1 = g1;
        this.j = j;
    }
    public String toString()
    {
        return g1 + ", j = " + j;
    }
}
public class Manager7

```

```

{
    public static void main(String[] args)
    {
        G g1 = new G(90);
        H h1 = new H(g1, 20);
        System.out.println(g1);
        System.out.println(h1);
    }
}

```

248. Program

```

package com.lara;
class I
{
    int x;
    I(int x)
    {
        this.x = x;
    }
    public String toString()
    {
        return "x = " + x;
    }
}
public class Manager8
{
    public static void main(String[] args)
    {

```

```

        I obj = new I(10);
        String s1 = "hello " + obj;
        System.out.println(s1);
    }
}

```

249. Program

```

package com.lara;
class K
{
    int i;
    K(int i)
    {
        this.i = i;
    }
    public String toString()
    {
        String s1 = super.toString();
        String s2 = "i = " + i;
        return s1 + " & " + s2;
    }
}
public class Manager9
{
    public static void main(String[] args)
    {
        K k1 = new K(10);
        System.out.println(k1);

        K k2 = new K(20);
        System.out.println(k2);
    }
}

```

```

    }
}

```

250. Program

```

package com.lara;
class L
{
    int i;
    L(int i)
    {
        this.i = i;
    }
    public String toString()
    {
        return "i = " + i;
    }
}
class M extends L
{
    int j;
    M(int i, int j)
    {
        super(i);
        this.j = j;
    }
    public String toString()
    {
        return super.toString() + ", j
= " + j;
    }
}

```

```

    }
}
public class Manager10
{
    public static void main(String[] args)
    {
        L obj1 = new L(90);
        M obj2 = new M(2, 40);
        System.out.println(obj1);
        System.out.println(obj2);
    }
}

```

251. Program

```

package com.lara;
class N
{
}
public class Manager11
{
    public static void main(String[] args)
    {
        N n1 = null;
        System.out.println(n1);
        String s1 = n1 + "abc";
        System.out.println(s1);
    }
}

```

252. Program

```

package com.rst;
class A
{
    int i;
}
public class Manager
{
    public static void main(String[] args)
    {
        A a1 = new A();
        a1.i = 10;
        A a2 = new A();
        a2.i = 10;
        A a3 = a1;
        System.out.println(a1 == a2);
        System.out.println(a2 == a3);
        System.out.println(a1 == a3);
    }
}

```

253. Program

```

package com.rst;
class B
{
    int i;
}
public class Manager1
{
    public static void main(String[] args)
    {

```

```

        B b1 = new B();
        B b2 = new B();
        B b3 = b1;
        b1.i = b2.i = 20;
        System.out.println(b1 == b2);
        System.out.println(b2 == b3);
        System.out.println(b3 == b1);
        System.out.println(b1.i ==
b2.i);
    }
}

```

254. Program

```

package com.rst;
class C
{
    int i;
    C(int i)
    {
        this.i = i;
    }
}
public class Manager2
{
    public static void main(String[] args)
    {
        C c1 = new C(10);

```

```

        C c2 = new C(10);
        System.out.println(c1 == c2);

        System.out.println(c1.equals(c2));
    }
}

```

255. Program

```

package com.rst;
class D
{
    int i;
    D(int i)
    {
        this.i = i;
    }
}
public class Manager3
{
    public static void main(String[] args)
    {
        D d1 = new D(90);
        D d2 = d1;
        System.out.println(d1 == d2);

        System.out.println(d1.equals(d2));
    }
}

```

256. Program

```

package com.rst;
class E
{
    int i;
    E(int i)
    {
        this.i = i;
    }
    public boolean equals(Object obj)
    {
        return this == obj;
    }
}
public class Manager4
{
    public static void main(String[] args)
    {
        E e1 = new E(90);
        E e2 = new E(90);
        E e3 = e1;
        System.out.println(e1.equals(e2));
        System.out.println(e2.equals(e3));
        System.out.println(e3.equals(e1));
    }
}

```

257. Program

```

package com.rst;
class F
{
    int i;

```

```

    F(int i)
    {
        this.i = i;
    }
    public boolean equals(Object obj)
    {
        F myObj = (F) obj;
        return this.i == myObj.i;
    }
}
public class Manager5
{
    public static void main(String[] args)
    {
        F f1 = new F(10);
        F f2 = new F(10);
        System.out.println(f1.equals(f2));
    }
}

```

258. Program

```

package com.rst;
class G
{
    int i, j;
    G(int i, int j)
    {
        this.i = i;
        this.j = j;
    }
}

```

```

    }
    public boolean equals(Object obj)
    {
        G myObj = (G) obj;
        return i == myObj.i && j ==
myObj.j;
    }
}
public class Manager6
{
    public static void main(String[] args)
    {
        G g1 = new G(1, 2);
        G g2 = new G(1, 2);
        System.out.println(g1.equals(g2));
    }
}

```

259. Program

```

package com.rst;
class H
{
    int i;
    double d;
    H(int i, double d)
    {
        this.i = i;
        this.d = d;
    }
}

```

```

    public String toString()
    {
        return "i = " + i + ", d = " + d;
    }

    public boolean equals(Object obj)
    {
        H myObj = (H) obj;
        boolean flag = (myObj.i == i
&& d ==
myObj.d);
        return flag;
    }
}
public class Manager7
{
    public static void main(String[] args)
    {
        H h1 = new H(9, 20.9);
        H h2 = new H(9, 20.9);
        System.out.println(h1.equals(h2));
    }
}

```

260. Program

```

package com.rst;
class L
{
    int i;
}

```

```

class M
{
    int i;
    public boolean equals(Object obj)
    {
        M myObj = (M) obj;
        return i == myObj.i;
    }
}

public class Manager8
{
    public static void main(String[] args)
    {
        L obj1 = new L();
        L obj2 = new L();
        M obj3 = new M();
        M obj4 = new M();
        obj1.i = 10;
        obj2.i = 10;
        obj3.i = 10;
        obj4.i = 10;
        System.out.println(obj1.equals(obj2)
);
        System.out.println("-----");
        System.out.println(obj3.equals(obj4)
);
        System.out.println("-----");
        System.out.println(obj1.equals(obj3)
);
        System.out.println("-----");

```

```

        System.out.println(obj3.equals(obj1)
);
    }
}

```

261. Program

```

package com.rst;
class P
{
    int i;
}
class Q
{
    int i;
    public boolean equals(Object obj)
    {
        if(!(obj instanceof Q))
        {
            return false;
        }
        Q myObj = (Q) obj;
        return i == myObj.i;
    }
}

public class Manager9
{
    262. public static void main(String[] args)
    {
        P p1 = new P();

```



```

        P p2 = new P();
        Q q1 = new Q();
        Q q2 = new Q();
        p1.i = 10;
        p2.i = 10;
        q1.i = 10;
        q2.i = 10;

        System.out.println(p1.equals(p2));
        System.out.println(q1.equals(q2));
        System.out.println(p1.equals(q2));
        System.out.println(q1.equals(p1));
    }
}

```

263. Program

```

package com.rst;
class R
{
    int i, j;
    double weight;
    R(int i, int j, double weight)
    {
        this.i = i;
        this.j = j;
        this.weight = weight;
    }

    public boolean equals(Object obj)
    {

```

```

        return (obj instanceof R &&
            i == ((R)obj).i &&
            j == ((R)obj).j &&
            weight == ((R)obj).weight);
    }
}

public class Manager10
{
    public static void main(String[] args)
    {
        R r1 = new R(10, 20, 9.98);
        R r2 = new R(10, 20, 9.98);
        System.out.println(r1.equals(r2));
    } }

```

264. Program

```

package com.rst;
class S
{
    int i;
}

public class Manager11
{
    public static void main(String[] args)
    {
        S s1 = new S();
        S s2 = new S();
        s1.i = s2.i = 10;
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s1.equals(s2));
    }
}

```

```

        System.out.println(s1.hashCode());
        System.out.println(s2.hashCode());
    }
}

```

265. Program

```

package com.rst;
class T
{
    int i;
    T(int i)
    {
        this.i = i;
    }
}
public class Manager12
{
    public static void main(String[] args)
    {
        T t1 = new T(90);
        T t2 = t1;
        System.out.println(t1);
        System.out.println(t2);
        System.out.println(t1.equals(t2));
        System.out.println(t1.hashCode());
        System.out.println(t2.hashCode());
    }
}

```

266. Program

```

package com.rst;
class U
{
    int i;
    U(int i)
    {
        this.i = i;
    }
    public String toString()
    {
        return "i = " + i;
    }
    public boolean equals(Object obj)
    {
        return (obj instanceof U && i
        == ((U)obj).i);
    }
    public int hashCode()
    {
        return i;
    }
}
public class Manager13
{
    public static void main(String[] args)
    {
        U u1 = new U(90);
        U u2 = new U(90);
        System.out.println(u1);
        System.out.println(u2);
    }
}

```

```

        System.out.println(u1.equals(u2));

        System.out.println(u1.hashCode());

        System.out.println(u2.hashCode());
    }
}

```

267. Program

```

package com.rst;

class V
{
    int i, j;
    V(int i, int j)
    {
        this.i = i;
        this.j = j;
    }
    public String toString()
    {
        return "i = " + i + ", j = " + j;
    }
    public boolean equals(Object obj)
    {
        return (obj instanceof V &&
                i == ((V)obj).i &&
                j == ((V)obj).j);
    }
    public int hashCode()
    {

```

```

        String s1 =
        Integer.toString(i);
        String s2 =
        Integer.toString(j);
        int hash = s1.hashCode();
        hash += s2.hashCode();
        return hash;
    }
}

public class Manager14
{
    public static void main(String[] args)
    {
        V v1 = new V(10, 40);
        V v2 = new V(10, 40);
        System.out.println(v1);
        System.out.println(v2);
        System.out.println(v1.equals(v2));
        System.out.println(v1.hashCode());
        System.out.println(v2.hashCode());
    }
}

```

268. Program

```

package com.rst;

class W
{
    int i;
    String s1;
    W(int i, String s1)

```

```

        {
            this.i = i;
            this.s1 = s1;
        }
        public int hashCode()
        {
            String str =
Integer.toString(i);
            int hash = str.hashCode();
            hash += s1.hashCode();
            return hash;
        }
    }
    public class Manager15
    {
        public static void main(String[] args)
        {
            W w1 = new W(10, "abc");
            W w2 = new W(10, "abc");
            System.out.println(w1.hashCode());
            System.out.println(w2.hashCode());
        }
    }

```

269. Program

```

package com.rst;
class X
{
    int i, j;
    X(int i, int j)

```

```

        {
            this.i = i;
            this.j = j;
        }
        public String toString()
        {
            return "i = " + i + ", j = " + j;
        }
        public boolean equals(Object obj)
        {
            return (obj instanceof X &&
i == ((X)obj).i
&&
j ==
((X)obj).j);
        }
        public int hashCode()
        {
            String s1 =
Integer.toString(i);
            String s2 =
Integer.toString(j);
            int hash = s1.hashCode();
            hash += s2.hashCode();
            return hash;
        }
    }
    public class Manager16
    {
        public static void main(String[] args)
        {

```

```

        X x1 = new X(9, 20);
        X x2 = new X(20, 9);
        System.out.println(x1);
        System.out.println(x2);

        System.out.println(x1.equals(x2));

        System.out.println(x1.hashCode());

        System.out.println(x2.hashCode());
    }
}

```

270. Program

```

package com.lara;
public class A implements Cloneable
{
    int i;
    public static void main(String[] args)
        throws
        CloneNotSupportedException
    {
        A a1 = new A();
        a1.i = 10;
        A a2 = (A)a1.clone();
        System.out.println(a2.i);
    }
}

```

271. Program

```

package com.lara;
public class B implements Cloneable
{
    int i;

    public static void main(String[] args)
        throws
        CloneNotSupportedException
    {
        B b1 = new B();
        b1.i = 10;
        B b2 = (B)b1.clone();
        System.out.println(b2.i);
        b2.i = 20;
        System.out.println(b1.i);
        b1.i = 30;
        System.out.println(b2.i);
        System.out.println(b1.i);
    }
}

```

272. Program

```

package com.lara;
public class C implements Cloneable
{
    int i;
    double d;
}

```

```

String s1;
Integer obj;
C(int i, double d, String s1, Integer
obj)
{
    this.i = i;
    this.d = d;
    this.s1 = s1;
    this.obj = obj;
}
public String toString()
{
    return "i = " + i + ", d = " + d
+ ", s1 = " + s1 + ", obj = " + obj;
}
public static void main(String[] args)
throws
CloneNotSupportedException
{
    C c1 = new C(20, 2.8, "abc",
45);
    System.out.println(c1);
    C c2 = (C) c1.clone();
    System.out.println(c2);
    c2.i = 40;
    c2.d = 200.909;
    c2.s1 = "cba";
    c2.obj = 400;
    System.out.println("---");
    System.out.println(c1);
    System.out.println(c2);

```

273. Program

```

package com.lara;
class D
{
    int i;
}
public class E implements Cloneable
{
    D d1;
    int j;

    public static void main(String[] args)
    throws
    CloneNotSupportedException
    {
        E e1 = new E();
        e1.d1 = new D();
        e1.d1.i = 10;
        e1.j = 20;

        E e2 = (E) e1.clone();
        System.out.println(e2.d1.i);
        System.out.println(e2.j);

        e2.d1.i = 100;
        e2.j = 200;
        System.out.println(e1.d1.i);

```

```

        System.out.println(e1.j);
    }
}

```

274. Program

```

package com.lara;

class F
{
    int i;
}

public class G implements Cloneable
{
    F fl;
    int j;

    protected Object clone()
    throws CloneNotSupportedException
    {
        Object obj = super.clone();
        G g1 = (G) obj;
        g1.fl = new F();
        g1.fl.i = fl.i;
        return g1;
    }

    public static void main(String[] args)
    throws
    CloneNotSupportedException
    {
        G g1 = new G();
        g1.fl = new F();
    }
}

```

```

g1.fl.i = 10;
g1.j = 20;
G g2 = (G) g1.clone();
g2.fl.i = 40;
g2.j = 50;
System.out.println(g1.fl.i);
System.out.println(g1.j);

```

275. Program

```

package com.lara;

public class M
{
    @Override
    protected void finalize() throws
    Throwable
    {
        System.out.println("from
        finalize");
    }

    public static void main(String[] args)
    {
        M m1 = new M();
        m1 = null;
        System.out.println("obj
        became abandoned");
        Runtime.getRuntime().gc();
        try
        {
            Thread.sleep(60*1000);
        }
    }
}

```

```

    }
    catch(InterruptedException
ex)
    {
        ex.printStackTrace();
    }
    System.out.println("end");
}
}

```

276. Program

```

package com.lara;
class A
{
}
public class Manager
{
    public static void main(String[] args)
    {
        A a1 = new A();
        A a2 = new A();
        System.out.println(a1 == a2);
        Class c1 = a1.getClass();
        Class c2 = a2.getClass();
        System.out.println(c1 == c2);
    }
}

```

277. Program

```

package com.lara;

```

```

class B
{
}
public class Manager1
{
    public static void main(String[] args)
    {
        B b1 = new B();
        B b2 = new B();
        B b3 = new B();
        B b4 = new B();
        Class c1 = b1.getClass();
        Class c2 = b2.getClass();
        Class c3 = b3.getClass();
        Class c4 = b4.getClass();
        System.out.println(c1 == c2);
        System.out.println(c2 == c3);
        System.out.println(c3 == c4);
        System.out.println(c4 == c1);
    }
}

```

278. Program

```

package com.lara;
class C
{
}

```



```

public class Manager2
{
    public static void main(String[] args)
    throws Exception
    {
        Class c1 =
Class.forName("com.lara.C");
        C obj1 = new C();
        Class c2 = obj1.getClass();
        System.out.println(c1 == c2);
    }
}

```

279. Program

```

package com.lara;
class D
{
}
public class Manager3
{
    public static void main(String[] args)
    {
        D d1 = new D();
        Class c1 = d1.getClass();
        Class c2 = null;
        try
        {
            c2 =
Class.forName("com.lara.D");

```

```

        }
    catch(ClassNotFoundException ex)
    {
        ex.printStackTrace();
    }
    Class c3 = D.class;
    System.out.println(c1 == c2);
    System.out.println(c2 == c3);
    System.out.println(c3 == c1);
}
}

```

280. Program

```

package com.lara;
import java.lang.reflect.Method;
class E
{
    public void test()
    {
        System.out.println("test");
    }
}
public class Manager4
{
    public static void main(String[] args)
    throws Exception
    {
        Class c1 =
Class.forName("com.lara.E");
        Object obj =
c1.newInstance();

```

```

        Method m1 =
c1.getDeclaredMethod("test");
        m1.invoke(obj);
        System.out.println("done");
    }
}

```

281. Program

```

package com.lara;
import java.lang.reflect.Method;
class F
{
    public void test1()
    {
        System.out.println("test1");
    }
    public void test2()
    {
        System.out.println("test2");
    }
}
public class Manager5
{
    public static void main(String[] args)
    throws Exception
    {
        Class c1 =

        Class.forName("com.lara.F");
        F f1 = (F) c1.newInstance();
        f1.test1();
    }
}

```

```

        f1.test2();
        System.out.println("-----");
        Object obj =
c1.newInstance();
        Method m1 =
c1.getDeclaredMethod("test1");
        Method m2 =
c1.getDeclaredMethod("test2");
        m1.invoke(obj);
        m2.invoke(obj);
    }
}

```

282. Program

```

package com.lara;
import java.lang.reflect.Method;
class G
{
    public void test(int i, String s1)
    {
        System.out.println("test:" +
i);
        System.out.println("test:" +
s1);
    }
}
public class Manager6
{
    public static void main(String[] args)
    {
    }
}

```

```

throws Exception
{
    Class c1 =

    Class.forName("com.lara.G");
    Object obj =
c1.newInstance();
    ((G)obj).test(10, "abc");

    System.out.println("=====");
    Method m1 =
c1.getDeclaredMethod("test", int.class,
String.class);
    m1.invoke(obj, 20, "xyz");
} }

```

283. Program

```

package com.lara;
import java.lang.reflect.Method;
import java.util.Scanner;
class H
{
    public void test1()
    {
        System.out.println("from H.test1");
    }
}
class I
{
    public void test2()

```

```

{
    System.out.println("from I.test2");
}

}

public class Manager7
{
    public static void main(String[] args)
        throws Exception
    {
        Scanner sc = new
Scanner(System.in);
        System.out.println("enter class
name");
        String className = sc.next();
        System.out.println("enter method
name");
        String methodName = sc.next();
        Class c1 =
Class.forName(className);
        Object obj = c1.newInstance();
        Method m1 =
c1.getDeclaredMethod(methodName);
        m1.invoke(obj);
        System.out.println("done");
    }
}

```

284. Program

```

package com.lara;
public class Manager8

```

```

{
    public static void main(String[] args)
        throws Throwable
    {
        Manager8 m1 = new
Manager8();
        m1.finalize();
        m1.finalize();
        m1.finalize();
        m1.finalize();
    }
}

```

285. program

```

package com.lara;
public class A
{
    public static void main(String[] args)
    {
        String s1="abc";
        System.out.println(s1);
    }
}

```

286. Program

```

package com.lara;
public class B
{
    public static void main(String[] args)
    {

```

```

String s1="abc";
String s2="abc";
System.out.println(s1);
System.out.println(s2);
}
}

```

287. Program

```

package com.lara;
public class C
{
    public static void main(String[] args)
    {
        String s1="abc";
        s1="xyz";
        System.out.println(s1);
    }
}

```

288. Program

```

package com.lara;
public class E
{
    public static void main(String[] args)
    {
        String s1="Lara";

        String s2="rst";
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s1.equals(s2));
        System.out.println(s1.toString());

```

```

System.out.println(s2.toString());
System.out.println(s1.hashCode());
System.out.println(s2.hashCode());
    }
}

```

289. Program

```

package com.lara;
public class G
{
    public static void main(String[] args)
    {
        String s1=new String("Lara");
        String s2=new String("Lara");
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s1.equals(s2));
        System.out.println(s2.equals(s1));
    }
}

```

290. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        String s1="Lara";
        String s2="Lara";
        String s3=new String("Lara");
        String s4=new String("Lara");
    }
}

```

```

System.out.println(s1);
System.out.println(s2);
System.out.println(s3);
System.out.println(s4);
System.out.println("_____");
System.out.println(s1==s2);
System.out.println(s1==s3);

        System.out.println(s1==s4);
        System.out.println(s3==s4);
        System.out.println("_____");
        System.out.println(s1.equals(s2));
        System.out.println(s1.equals(s3));
        System.out.println(s1.equals(s4));
        System.out.println("_____");
        System.out.println(s1.hashCode()==
s2.hashCode());
        System.out.println(s1.hashCode()==
s3.hashCode());
        System.out.println(s1.hashCode()==
s4.hashCode());
        System.out.println("___Done___");
    }
}

```

291. Program

```

package com.lara;
public class J
{
    public static void main(String[] args)
    {
    }
}

```

```
String s1="Lara";
System.out.println(s1);
s1=s1+"Rst";
System.out.println(s1);
}
```

```
}
```

292. Program

```
package com.lara;
public class K
{
    public static void main(String[] args)
    {
        String s1="java";
        String s2="ja"+"va";
        System.out.println(s1==s2);
    }
}
```

293. Program

```
package com.lara;
public class L
{
    public static void main(String[] args)
    {
        String s1="java";
        String s2="ja";
        String s3=s2+"va";
        System.out.println(s1==s3);
    }
}
```

294. Program

```
package com.lara;
public class M
{
    public static void main(String[] args)
    {
        String s1="java";
        String s2="ja";
        String s3="va";
        String s4=s2+s3;
        System.out.println(s1==s4);
    }
}
```

295. Program

```
package com.lara;
public class N
{
    public static void main(String[] args)
    {
        String s1=null;
        System.out.println(s1);
        System.out.println(s1);
        s1=s1+s1+null;
        System.out.println(s1);
    }
}
```

296. Program

```
package com.lara;
public class O
```

```

{
public static void main(String[] args)
{
    System.out.println(2+4);
    System.out.println(2+4+"lara");
    System.out.println(2+"lara"+4);
    System.out.println("lara"+2+4);
    System.out.println("lara"+2+null);
    System.out.println(null+"abc");
}
}

```

297. Program

```

package com.lara;
public class O
{
public static void main(String[] args)
{
    System.out.println(2+4);
    System.out.println(2+4+"lara");
    System.out.println(2+"lara"+4);
    System.out.println("lara"+2+4);
    System.out.println("lara"+2+null);
    System.out.println(null+"abc");
}
}

```

298. Program

```

package com.lara;
public class Q
{

```

```

public static void main(String[] args)
{
    String
    s1="abc";System.out.println(s1);s1.concat("
    Rst");
    System.out.println(s1);
}
}

```

299. Program

```

package com.lara;
public class R
{
    public static void main(String[] args)
    {
        String s1="Lara";
        s1=s1.concat("Rst");
        System.out.println(s1);
    }
}

```

300. Program

```

package com.lara;
public class S
{
    public static void main(String[] args)
    {
        String s1="ja";
        String s2="va";
        String s3=s1.concat(s2);
        String s4="java";

```

```

        System.out.println(s3==s4);
    }
}

```

301. Program

```

package com.lara;
public class T
{
    public static void main(String[] args)
    {
        String s1="ja"+"va";
        String s2="java";
        System.out.println(s1==s2);
    }
}

```

302. Program

```

package com.lara;
public class U
{
    public static void main(String[] args)
    {
        String s1="ja".concat("va");
        String s2="java";
        System.out.println(s1==s2);
    }
}

```

303. Program

```

package com.lara;
public class V
{

```

```

    public static void main(String[] args)
    {
        String s1="abc";
        System.out.println(s1.length());
    }
}

```

304. Program

```

package com.lara;
public class W
{
    public static void main(String[] args)
    {
        String s1=null;
        System.out.println(s1.length());
    }
}

```

305. Program

```

package com.lara;
public class X
{
    public static void main(String[] args)
    {
        String s1="null";
        System.out.println(s1.length());
    }
}

```

306. Program

```

package com.lara;

```



```

public class Y
{
    public static void main(String[] args)
    {
        String s1=" abc 123 ";

        s1.trim();

        System.out.println(s1.length());
    }
}

```

307.Program

```

package com.lara;

public class Z
{
    public static void main(String[] args)
    {
        String s1=" abc 123 ";
        s1=s1.trim();
        System.out.println(s1.length());
    }
}

```

308.Program

```

package com.lara;

public class M1
{
    public static void main(String[] args)
    {
        String s1="abc123xyba"
        System.out.println(s1.charAt(4));
    }
}

```

```

        System.out.println(s1.charAt(6));
        System.out.println(s1.charAt(2));
        System.out.println(s1.charAt(15));
    }
}

```

309.Program

```

package com.lara;

public class M2
{
    public static void main(String[] args)
    {
        String s1="a1b2c3a1b4";
        System.out.println(s1.indexOf('a'));
        System.out.println(s1.indexOf('a',4));
        System.out.println(s1.lastIndexOf('b'));
        System.out.println(s1.lastIndexOf('b',4));
    }
}

```

310.Program

```

package com.lara;

public class M3
{
    public static void main(String[] args)
    {
        String s1="a1b2c3a1b4";
        System.out.println(s1.indexOf('z'));
        System.out.println(s1.lastIndexOf('p'));
    }
}

```

```

}
}

```

311. Program

```

package com.lara;

public class M5
{
    public static void main(String[] args)
    {
        String s1="abc;123;xyz;hello";
        String x[]=s1.split(";");
        for(String obj:x)
        {
            System.out.println(obj);
        }
    }
}

```

312. Program

```

package com.lara;

public class M6
{
    public static void main(String[] args)
    {
        String s1="lara tech";
        String s2=s1.substring(0,4);
        String s3=s1.substring(5,9);
        String s4=s1.substring(5);
        System.out.println(s2);
        System.out.println(s3);
        System.out.println(s4);
    }
}

```

```

}
}

```

313. Program

```

package com.lara;

public class M7
{
    public static void main(String[] args)
    {
        String s1="Abcxyz123";
        System.out.println(s1.toUpperCase());
        System.out.println(s1.toLowerCase());
        System.out.println(s1);
    }
}

```

314. Program

```

package com.lara;

public class M8
{
    public static void main(String[] args)
    {
        String s1="lara tech";
        System.out.println(s1.startsWith("lara"));
        System.out.println(s1.endsWith("tech"));
        System.out.println(s1.startsWith("abc"));
        System.out.println(s1.endsWith("abc"));
    }
}

```

315. Program

```

package com.lara;

public class A
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        sb.append("abc\n");
        sb.append("xyz\n");
        sb.append("hello");
        System.out.println(sb);
    }
}

```

316. Program

```

package com.lara;

public class B
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        sb.append("abc\n");
        sb.append("abc\n");
        sb.append("abc\n");
        sb.append("abc");
        System.out.println(sb);
    }
}

```

317. Program

```

package com.lara;

```

```

public class C
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer("abc");
        StringBuffer sb1=new StringBuffer("abc");
        System.out.println(sb.toString());
        System.out.println(sb1.toString());
        System.out.println(sb==sb1);
        System.out.println(sb.hashCode());
        System.out.println(sb1.hashCode());
        System.out.println(sb.equals(sb1));
    }
}

```

318. Program

```

package com.lara;

public class D
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        sb.append("abcabcabcabcabc");
        System.out.println(sb);
        System.out.println(sb.capacity());
        System.out.println(sb.length());
        sb.trimToSize();
        System.out.println(sb.capacity());
        System.out.println(sb.length());
    }
}

```

319. Program

```

package com.lara;

public class E
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        System.out.println(sb.capacity());
        sb.append("abcabcabcabcabcabcabc");
        System.out.println(sb.length());
        System.out.println(sb.capacity());
        System.out.println("-----");
        sb.append("abcabcabcabcabcabcabc");
        System.out.println(sb.length());
        System.out.println(sb.capacity());
        System.out.println("-----");
        sb.trimToSize();
        System.out.println(sb.capacity());
        System.out.println(sb.length());
    }
}

```

320. Program

```

package com.lara;

public class F
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        sb.append("abc");

```

```

        sb.append("xyz");
        sb.append("rst");
        System.out.println(sb);
        sb.reverse();
        System.out.println(sb);
    }
}

```

321. Program

```

package com.lara;

public class G
{
    public static void main(String[] args)
    {
        StringBuffer sb=new StringBuffer();
        sb.append("abc");
        sb.append("xyz");
        sb.append("rst");
        System.out.println(sb);
        sb.delete(3, 6);
        System.out.println(sb);
    }
}

```

322. Program

```

package com.lara;

public class H
{
    public static void main(String[] args)
    {

```

```

StringBuilder sb=new StringBuilder();
sb.append("abc\n");
sb.append("xyz\n");
sb.append("rst");
System.out.println(sb);
}
}

```

323.Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class A
{
    public static void main(String[] args)
    {
        String s1="abab89abc1abc1s2";
        Pattern p1=Pattern.compile("abc");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

324.Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class B
{
    public static void main(String[] args)
    {
        String s1="abcabaabc";
        Pattern p1=Pattern.compile("[ab]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

325.Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class C
{
    public static void main(String[] args)
    {
        String s1="abcxyz1pqr2km9rt";
        Pattern
p1=Pattern.compile("[abkpq]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())

```

```

        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

326. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class D
{
    public static void main(String[] args)
    {
        String s1="abcxyz1pqr2km9rt";
        Pattern p1=Pattern.compile("[a-p]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

327. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class E

```

```

{
    public static void main(String[] args)
    {
        String s1="abcxyz1pqr2km9rt";
        Pattern p1=
Pattern.compile("[123456]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

328. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class F
{
    public static void main(String[] args)
    {
        String
s1="abcxyz1pqr2km9rt";
        Pattern p1=
Pattern.compile("[1-9]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {

```

```

        System.out.println(m1.start()+":"+m
1.group());
    }

}

```

329. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class G
{
    public static void main(String[] args)
    {
        String
s1="abcxyz1pqr2km9rt";
        Pattern p1=
Pattern.compile("[a-ep-z1-35-9]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

330. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class G
{
    public static void main(String[] args)
    {
        String
s1="abcxyz1pqr2km9rt";
        Pattern p1=
Pattern.compile("[a-ep-z1-35-9]");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

331. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class I
{
    public static void main(String[] args)

```

```

        {
            String
s1="abcxyz1pqr2km9rt";
            Pattern p1=
Pattern.compile("\\d");
            Matcher m1=p1.matcher(s1);
            while(m1.find())
            {
                System.out.println(m1.start()+":"+m
1.group());
            }
        }
    }
}

```

332. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class J
{
    public static void main(String[] args)
    {
        String
s1="abcxyz1pqr2km9rt";
        Pattern p1=
Pattern.compile("\\d+");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {

```

```

            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

333. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class K
{
    public static void main(String[] args)
    {
        String s1="a-b$6AB%1jbc";
        Pattern p1=
Pattern.compile("\\w");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

334. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class L

```



```

{
    public static void main(String[] args)
    {
        String s1="abc xyz 123";
        Pattern p1=
Pattern.compile("\\s");
        Matcher m1=p1.matcher(s1);
        while(m1.find())
        {
            System.out.println(m1.start()+":"+m
1.group());
        }
    }
}

```

335. Program

```

package com.lara;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class M
{
    public static void main(String[] args)
    {
        String s1="a1b2c3d4";
        String x[]=s1.split("\\d");
        for(String str:x)
        {

            System.out.println(str);
        }
    }
}

```

```

}

```

336. Program

```

package com.lara;
public class N
{
    public static void main(String[] args)
    {
        String s1="a1b2c3d4";
        String x[]=s1.split("[0-9]");
        for(String str:x)
        {

            System.out.println(str);
        }
    }
}

```

337. Program

```

package com.lara;
public class O
{
    public static void main(String[] args)
    {
        String s1="a1b2c3d4";
        String x[]=s1.split("2");
        for(String str:x)
        {

            System.out.println(str);
        }
    }
}

```

```

    }
}
}

```

338. Program

```

package com.lara;
import java.util.StringTokenizer;
public class P
{
    public static void main(String[] args)
    {
        StringTokenizer st=new
StringTokenizer("abc:xyz:hello",":");
        while (st.hasMoreElements())
        {
            System.out.println(st.nextElement());
        }
    }
}

```

339. Program

```

package com.lara;
import java.util.StringTokenizer;
public class Q
{
    public static void main(String[] args)
    {
        StringTokenizer st=new
StringTokenizer("a1b2c3d4","\d");
        while (st.hasMoreElements())

```

```

    {
        System.out.println(st.nextElement());
    }
}
}

```

340. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("%s","Hello");
    }
}

```

341. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("(%s)","Hello");
    }
}

```

342. Program

```

package com.lara;
public class I

```

```

{
    public static void main(String[] args)
    {

        System.out.printf("(%s)","Hello");
    }
}

```

343. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("%s to only
        %S","hello","ladies");
    }
}

```

344. Program

```

package com.lara;
public class I

{
    public static void main(String[] args)
    {
        System.out.printf("%d + %d= %d",1,2,3);
    }
}

```

345. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("%2$d + %1$d =
        %3$d",1,2,3);
    }
}

```

346. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("%2$d +
        %1$d = %3$d",1,2,3);
    }
}

```

347. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("(%10d)",100);
    }
}

```

}

348. Program

package com.lara;

public class I

{

public static void main(String[] args)

{

System.out.printf("(%-

10d)",23);

}

}

349. Program

package com.lara;

public class I

{

public static void main(String[] args)

{

System.out.printf("<%010d>",23);

}

}

350. Program

package com.lara;

public class I

{

public static void main(String[] args)

{

System.out.printf("<%0-

10d>",23);

}

}

351. Program

package com.lara;

public class I

{

public static void main(String[] args)

{

System.out.printf("%0,10d)",100000

);

}

}

352. Program

package com.lara;

public class I

{

public static void main(String[] args)

{

System.out.printf("<%-

+,10d>",1000);

}

}

353. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("%-10d",-100);
    }
}

```

354. Program

```

package com.lara;
public class I
{
    public static void main(String[] args)
    {
        System.out.printf("PI value
is: %f",Math.PI);
    }
}

```

355. Program

```

package com.lara;
import java.util.Date;
public class A
{
    public static void main(String[] args)
    {

```

```

        Date d1=new Date();
        System.out.println(d1);

```

```

    }

```

```

}

```

356. Program

```

package com.lara;
import java.util.Date;
public class B
{
    public static void main(String[] args)
    {
        Date d1=new Date(0);
        System.out.println(d1);
    }
}

```

357. Program

```

package com.lara;
import java.util.Date;

public class B
{
    public static void main(String[] args)
    {
        Date d1=new Date(1000*60*60);
        System.out.println(d1);
    }
}

```

358. Program

```

package com.lara;
import java.util.Date;

public class B
{
    public static void main(String[] args)
    {
        Date d1=new
Date(24*1000*60*60);
        System.out.println(d1);
    }
}

```

359. Program

```

package com.lara;
import java.util.Date;
public class C
{
    public static void main(String[] args)
    {
        Date d1=new
Date(24*1000*60*60);
        System.out.println(d1);
        long millis=d1.getTime();
        System.out.println(millis);
    }
}

```

360. Program

```

package com.lara;
import java.util.Date;
public class C
{
    public static void main(String[] args)
    {
        Date d1=new Date(-
24*1000*60*60);
        System.out.println(d1);
        long millis=d1.getTime();
        System.out.println(millis);
    }
}

```

361. Program

```

package com.lara;
import java.util.Date;
public class C
{
    public static void main(String[] args)
    {
        Date d1=new Date();
        System.out.println(d1);
        long millis=d1.getTime();
        System.out.println(millis);
    }
}

```

```

    }
}

```

362. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

363. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.DATE, 2);
        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

```

    }
}

```

364. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.DATE, -2);
        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

365. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.MONTH, 2);
        Date d1=c1.getTime();
    }
}

```

```

        System.out.println(d1);
    }
}

```

366. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.MONTH, -2);
        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

367. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.YEAR, 2);
    }
}

```

```

        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

368. Program

```

package com.lara;
import java.util.Calendar;
import java.util.Date;
public class D
{
    public static void main(String[] args)
    {
        Calendar c1=Calendar.getInstance();
        c1.add(Calendar.YEAR, -2);
        Date d1=c1.getTime();
        System.out.println(d1);
    }
}

```

369. Program

```

package com.lara;
import java.text.DateFormat;
import java.util.Date;
public class E
{
    public static void main(String[] args)
    {
        Date d1=new Date();
    }
}

```



```

        System.out.println(d1);
        DateFormat
df=DateFormat.getDateInstance(DateFormat
.SHORT);
        String s1=df.format(d1);
        System.out.println(s1);
    }
}

```

370. Program

```

package com.lara;
import java.text.DateFormat;
import java.util.Date;
public class E
{
    public static void main(String[] args)
    {
        Date d1=new Date();
        System.out.println(d1);
        DateFormat
df=DateFormat.getDateInstance(DateFormat
.MEDIUM);
        String s1=df.format(d1);
        System.out.println(s1);
    }
}

```

371. Program

```

package com.lara;
import java.text.DateFormat;

```

```

import java.util.Date;
public class E
{
    public static void main(String[] args)
    {
        Date d1=new Date();
        System.out.println(d1);
        DateFormat
df=DateFormat.getDateInstance(DateFormat
.FULL);
        String s1=df.format(d1);
        System.out.println(s1);
    }
}

```

372. Program

```

package com.lara;
import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;
public class F
{
    public static void main(String[] args)
    {
        Date d1=new Date();
        System.out.println(d1);
        DateFormat
df=DateFormat.getDateInstance(DateFormat
.SHORT, new Locale("it"));
        String s1=df.format(d1);
    }
}

```

```

        System.out.println(s1);
    }
}

```

373. Program

```

package com.lara;
import java.text.NumberFormat;
import java.util.Locale;
public class G
{
    public static void main(String[] args)
    {
        double d1=102929.8789789;
        System.out.println(d1);
        NumberFormat
        nf1=NumberFormat.getInstance(Locale.ITA
        LY);
        String s1=nf1.format(d1);
        System.out.println(s1);
    }
}

```

374. Program

```

package com.lara;
import java.text.NumberFormat;
import java.util.Locale;
public class G
{
    public static void main(String[] args)

```

```

    {
        double d1=102929.8789789;
        System.out.println(d1);
        NumberFormat
        nf1=NumberFormat.getInstance();
        String s1=nf1.format(d1);
        System.out.println(s1);
    }
}

```

375. Program

```

package com.lara;
import java.text.NumberFormat;
import java.util.Locale;
public class G
{
    public static void main(String[] args)
    {
        double d1=102929.8789789;
        System.out.println(d1);
        NumberFormat
        nf1=NumberFormat.getCurrencyInstance();
        String s1=nf1.format(d1);
        System.out.println(s1);
    }
}

```

376. Program

```
package com.lara;  
import java.text.NumberFormat;  
import java.util.Locale;  
public class G  
{  
    public static void main(String[] args)  
    {  
        double d1=102929.8789789;  
        System.out.println(d1);  
        NumberFormat  
nf1=NumberFormat.getCurrencyInstance(L  
ocale.UK);  
        String s1=nf1.format(d1);  
        System.out.println(s1);  
    }  
}
```

SimpleDateFormat Pattern Fields

The formatter supports many different fields you can use to create patterns. ASCII letters are reserved as pattern letters (unless enclosed in quotes), defined as the following:

Symbol	Meaning	Presentation	Example
G	era designator	(Text)	AD
y	year	(Number)	1996
M	month in year	(Text / Number)	July / 07
d	day in month	(Number)	10
k	hour in day	(Number)	24
H	hour in day, 0-based	(Number)	0
m	minute in hour	(Number)	30
s	second in minute	(Number)	55
S	millisecond	(Number)	978
E	day in week	(Text)	Tuesday
D	day in year	(Number)	189
F	day of week in month	(Number)	2 (2nd Wed in July)
w	week in year	(Number)	27
W	week in month	(Number)	2
a	am/pm marker	(Text)	PM
h	hour in am/pm	(Number)	12
K	hour in am/pm, 0-based	(Number)	0
z	time zone	(Text)	PST
'	escape for text		
"	single quote		'

377. Program

```

import java.text.*;
import java.util.*;

public class SimpleDateFormatDemo
{
    public static void main(String args[])
    {
        Date date = new Date();
        SimpleDateFormat sdf;
        sdf = new
SimpleDateFormat("hh:mm:ss");
        System.out.println(sdf.format(date));
        sdf = new SimpleDateFormat("dd
MMM yyyy hh:mm:ss zzz");
        System.out.println(sdf.format(date));
        sdf = new SimpleDateFormat("E
MMM dd yyyy");
        System.out.println(sdf.format(date));
    }
}

```

378. Program

```

import java.text.*;
import java.util.*;

public class DateFormat
{
    public static void main(String args[])

```

```

{
    String s;
    Format formatter;
    Date date = new Date();
    // 01/09/02
    formatter = new
SimpleDateFormat("MM/dd/yy");
    s = formatter.format(date);
    System.out.println(s);
    // 01/09/02
    formatter = new
SimpleDateFormat("dd/MM/yy");
    s = formatter.format(date);
    System.out.println(s);
    // 29-Jan-02
    formatter = new
SimpleDateFormat("dd-MMM-yy");
    s = formatter.format(date);
    System.out.println(s);
    // 2002.01.29.08.36.33
    formatter = new
SimpleDateFormat("yyyy.MM.dd.H
H.mm.ss");
    s = formatter.format(date);
    System.out.println(s);
    // Tue, 09 Jan 2002 22:14:02 -0500
    formatter = new
SimpleDateFormat("E, dd MMM
yyyy HH:mm:ss Z");
    s = formatter.format(date);
    System.out.println(s);
}

```

```
        formatter = new  
        SimpleDateFormat("EEEE, dd  
        MMMM yyyy HH:mm:ss zzzz");  
        s = formatter.format(date);  
        System.out.println(s);  
    }  
}
```

LARA

SCJP EXAMPLES

1. Program

A developer is creating a class Book that needs to access class Paper.

The Paper class is deployed in a JAR named myLib.jar. Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)

- A. The JAR file is located at \$JAVA_HOME/jre/classes/myLib.jar.
- B. The JAR file is located at \$JAVA_HOME/jre/lib/ext/myLib.jar.
- C. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar/Paper.class.
- D. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar.
- E. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -cp /foo/myLib.jar/Paper Book.java.
- F. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -d /foo/myLib.jar Book.java.
- G. The JAR file is located at /foo/myLib.jar and the Book class is

compiled using javac -classpath /foo/myLib.jar Book.java.

Answer: BDG

2. Program

Given:

1. package com.company.application;
- 2.
3. public class MainClass {
4. public static void main(String[] args) { }
5. }

And MainClass exists in the /apps/com/company/application directory.

Assume the CLASSPATH environment variable is set to "." (current directory). Which two java commands entered at the command line will run MainClass? (Choose two.)

- A. java MainClass if run from the /apps directory
- B. java com.company.application.MainClass if run from the /apps directory
- C. java -classpath /apps com.company.application.MainClass if run from any directory
- D. java -classpath . MainClass if run from the /apps/com/company/application directory

E. `java -classpath`

`/apps/com/company/application:. MainClass`
if run

from the `/apps` directory

F. `java com.company.application.MainClass`
if run from the

`/apps/com/company/application` directory

Answer: BC

3. Program

Given a correctly compiled class whose source code is:

1. `package com.sun.sjcp;`
2. `public class Commander {`
3. `public static void main(String[] args) {`
4. `// more code here`
5. `}`
6. `}`

Assume that the class file is located in `/foo/com/sun/sjcp/`, the current directory is `/foo/`, and that the classpath contains “.” (current directory).

Which command line correctly runs Commander?

- A. `java Commander`
- B. `java com. sim. sjcp.Commander`
- C. `java com/sun/sjcp/Commander`
- D. `java -cp com.sun.sjcp Commander`
- E. `java -cp com/sun/sjcp Commander`

Answer: B

4. Program

A UNIX user named Bob wants to replace his chess program with a new one, but he is not sure where the old one is installed. Bob is currently able to run a Java chess program starting from his home

directory `/home/bob` using the command:

`java -classpath /test:/home/bob/downloads/*`
`.jar games.Chess`

Bob's CLASSPATH is set (at login time) to:
`/usr/lib:/home/bob/classes:/opt/java/lib:/opt/java/lib/* .jar`

What is a possible location for the `Chess.class` file?

- A. `/test/Chess.class`
- B. `/home/bob/Chess.class`
- C. `/test/games/Chess.class`
- D. `/usr/lib/games/Chess.class`
- E. `/home/bob/games/Chess.class`
- F. inside jarfile `/opt/java/lib/Games.jar` (with a correct manifest)
- G. inside jarfile `/home/bob/downloads/Games.jar` (with a correct manifest)

Answer: C

5. Program

A class `games.cards.Poker` is correctly defined in the jar file `Poker.jar`.

A user wants to execute the main method of `Poker` on a UNIX system using the command:

```
java games.cards.Poker
```

What allows the user to do this?

- A. put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java`
- B. put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java/*.jar`
- C. Put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java/Poker.jar`
- D. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java`
- E. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java/*.jar`
- F. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java/Poker.jar`

Answer: C

6. Program

Given:

```
11. public class Ball {
12.     public enum Color { RED, GREEN,
13.         BLUE };
14.     public void foo() {
15.         // insert code here
16.     }
17. }
```

Which code inserted at line 14 causes the `foo` method to print RED, GREEN, and BLUE?

- A. `for(Color c : Color.values())`
- B. `for(Color c = RED; c <= BLUE; c++)`
- C. `for(Color c; c.hasNext() ; c.next())`
- D. `for(Color c = Color[0]; c <= Color[2]; c++)`
- E. `for(Color c = Color.RED; c <= Color.BLUE; c++)`

Answer: A

7. Program

Given:

```
11. public class Test {
12.     public enum Dogs {collie, harrier,
13.         shepherd};
14.     public static void main(String [] args) {
15.         Dogs myDog = Dogs.shepherd;
16.         switch (myDog) {
17.             case collie:
```

```

17. System.out.print("collie ");
18. case default:
19. System.out.print("retriever ");
20. case harrier: 21.
System.out.print("harrier ");
22. }
23. }
24. }

```

‘What is the result?

- A. harrier
- B. shepherd
- C. retriever
- D. Compilation fails.
- E. retriever harrier
- F. An exception is thrown at runtime.

Answer: D

8. Program

Given:

```

12. public class Test {
13. public enum Dogs {collie, harrier};
14. public static void main(String [] args) {
15. Dogs myDog = Dogs.collie;
16. switch (myDog) {
17. case collie:
18. System.out.print("collie ");
19. case harrier:
20. System.out.print("harrier ");
21. }
22. }

```

```

23. }

```

What is the result?

- A. collie
- B. harrier
- C. Compilation fails.
- D. collie harrier
- E. An exception is thrown at runtime.

Answer: D

9. Program

___ Given:

```

10. public class Fabric
11. public enum Color {
12. RED(0xff0000), GREEN(0x00ff00),
BLUE(0x0000ff);
13. private final int rgb;
14. Color( int rgb) { this.rgb = rgb; }
15. public int getRGB() { return rgb; }
16. };
17. public static void main( String[] argv) {
18. // insert code here
19. }
20. }

```

Which two code fragments, inserted independently at line 18, allow the Fabric class to compile? (Choose two.)

- A. Color skyColor = BLUE;
- B. Color treeColor = Color.GREEN;
- C. Color purple = new Color(0xff00ff);
- D. if(RED.getRGB() < BLUE.getRGB()) {}

E. `Color purple = Color.BLUE + Color.RED;`
 F. `if(Color.RED.ordinal() < Color.BLUE.ordinal()) {}`

Answer: BF

10. Program

Given:

```
10. class Nav{
11. public enum Direction { NORTH,
    SOUTH, EAST, WEST }
12. }
13. public class Sprite{
14. // insert code here
15. }
```

Which code, inserted at line 14, allows the Sprite class to compile?

- A. `Direction d = NORTH;`
- B. `Nav.Direction d = NORTH;`
- C. `Direction d = Direction.NORTH;`
- D. `Nav.Direction d = Nav.Direction.NORTH;`

Answer: D

11. Program

Given:

```
11. public enum Title {
12. MR("Mr."), MRS("Mrs."), MS("Ms.");
13. private final String title;
```

```
14. private Title(String t) { title = t; }
15. public String format(String last, String
    first) {
16. return title + " " + first + " " + last;
17. }
18. }
19. public static void main(String[] args) {
20.
    System.out.println(Title.MR.format("Doe",
        "John"));
21. }
```

What is the result?

- A. *Mr. John Doe*
- B. *An exception is thrown at runtime.*
- C. *Compilation fails because of an error in line 12.*
- D. *Compilation fails because of an error in line 15.*
- E. *Compilation fails because of an error in line 20.*

Answer: A

12. Program

Given:

```
10. package com.sun.scjp;
11. public class Geodetics {
12. public static final double DIAMETER =
    12756.32; // kilometers
13. }
```

Which two correctly access the DIAMETER member of the Geodetics

class? (Choose two.)

- A. `import com.sun.scjp.Geodetics;`
`public class TerraCarta {`
`public double halfway()`
`{ return Geodetics.DIAMETER/2.0; } }`
- B. `import static com.sun.scjp.Geodetics;`
`public class TerraCarta {`
`public double halfway() { return`
`DIAMETER/2.0; } }`
- C. `import static com.sun.scjp.Geodetics. *;`
`public class TerraCarta {`
`public double halfway() { return`
`DIAMETER/2.0; } }`
- D. `package com.sun.scjp;`
`public class TerraCarta {`
`public double halfway() { return`
`DIAMETER/2.0; } }`

Answer: AC

13. Program

Given:

1. `package sun.scjp;`
2. `public enum Color { RED, GREEN, BLUE }`
1. `package sun.beta;`
2. `// insert code here`
3. `public class Beta {`
4. `Color g = GREEN;`
5. `public static void main(String[] argv)`

6. `{ System.out.println(GREEN); }`
7. `}`

The class Beta and the enum Color are in different packages.

Which two code fragments, inserted individually at line 2 of the Beta declaration, will allow this code to compile? (Choose two.)

- A. `import sun.scjp.Color.*;`
- B. `import static sun.scjp.Color.*;`
- C. `import sun.scjp.Color; import static sun.scjp.Color.*;`
- D. `import sun.scjp.*; import static sun.scjp.Color.*;`
- E. `import sun.scjp.Color; import static sun.scjp.Color.GREEN;`

Answer: CE

14. Program

__Given a class Repetition:

1. `package utils;`
2.
3. `public class Repetition {`
4. `public static String twice(String s) {`
`return s + s; }`
5. `}`

and given another class Demo:

1. *// insert code here*

```

2.
3. public class Demo {
4. public static void main(String[] args) {
5. System.out.println(twice("pizza"));
6. }
7. }

```

Which code should be inserted at line 1 of Demo.java to compile and run Demo to print "pizzapizza"?

- A. import utils.*;
- B. static import utils.*;
- C. import utils.Repetition.*;
- D. static import utils.Repetition.*;
- E. import utils.Repetition.twice();
- F. import static utils.Repetition.twice;
- G. static import utils.Repetition.twice;

Answer:F

15. Program

Given:

```

10. class Line {
11. public static class Point { }
12. }
13.
14. class Triangle {
15. // insert code here
16. }

```

Which code, inserted at line 15, creates an instance of the Point class defined in Line?

- A. Point p = new Point();

B. Line.Point p = new Line.Point();

C. The Point class cannot be instantiated at line 15.

D. Line l = new Line() ; l.Point p = new l.Point();

Answer: B

16. Program

Given:

```

10. class Line {
11. public class Point { public int x,y; }
12. public Point getPoint() { return new
Point(); }
13. }
14. class Triangle {
15. public Triangle() {
16. // insert code here
17. }18. }

```

Which code, inserted at line 16, correctly retrieves a local instance of a Point object?

- A. Point p = Line.getPoint();
- B. Line.Point p = Line.getPoint();
- C. Point p = (new Line()).getPoint();
- D. Line.Point p = (new Line()).getPoint();

Answer: D

17. Program

_Given:

```

1. package geometry;
2. public class Hypotenuse {
3.     public InnerTriangle it = new
   InnerTriangle();
4.     class InnerTriangle {
5.         public int base;
6.         public int height;
7.     }
8. }

```

Which is true about the class of an object that can reference the variable base?

- A. It can be any class.*
- B. No class has access to base.*
- C. The class must belong to the geometry package.*
- D. The class must be a subclass of the class Hypotenuse.*

Answer: C

18. Program

Click the Exhibit button.

```

10. interface Foo {
11.     int bar();
12. }
13.
14. public class Beta {
15.
16.     class A implements Foo {

```

```

17.         public int bar() { return 1; }
18.     }
19.
20.     public int fubar( Foo foo) { return
   foo.bar(); }
21.
22.     public void testFoo() {
23.
24.     class A implements Foo {
25.         public int bar() { return 2; }
26.     }
27.
28.     System.out.println( fubar( new A()));
29. }
30.
31. public static void main( String[] argv) {
32.     new Beta().testFoo();
33. }
34. }

```

Which three statements are true? (Choose three.)

- A. Compilation fails.*
- B. The code compiles and the output is 2.*
- C. If lines 16, 17 and 18 were removed, compilation would fail.*
- D. If lines 24, 25 and 26 were removed, compilation would fail.*
- E. If lines 16, 17 and 18 were removed, the code would compile and the output would be 2.*

F. If lines 24, 25 and 26 were removed, the code would compile and the output would be 1.

Answer: BEF

19. Program

___Given:

```
1. interface TestA { String toString(); }
2. public class Test {
3. public static void main(String[] args) {
4. System.out.println(new TestA() {
5. public String toString() { return "test"; }
6. });
7. }
8. }
```

What is the result?

- A. test*
- B. null*
- C. An exception is thrown at runtime.*
- D. Compilation fails because of an error in line 1.*
- E. Compilation fails because of an error in line 4.*
- F. Compilation fails because of an error in line 5.*

Answer: A

20. Program

___Given:

```
10. interface Foo { int bar(); }
11. public class Sprite {
12. public int fubar( Foo foo) { return
foo.bar(); }
13. public void testFoo() {
14. fubar(
15. //insert code here
16.);
17. }
18. }
```

Which code, inserted at line 15, allows the class Sprite to compile?

- A. Foo { public int bar() { return 1; } }*
- B. new Foo { public int bar() { return 1; } }*
- C. new Foo() { public int bar() { return 1; } }*
- D. new class Foo { public int bar() { return 1; } }*

Answer: C

21. Program

Given:

```
11. class Converter {
12. public static void main(String[] args) {
13. Integer i = args[0];
14. int j = 12;
15. System.out.println("It is " + (j==i) +
"that j==i.");
```


16. }

17. }

What is the result when the programmer attempts to compile the code and run it with the command `java Converter 12?`

- A. It is true that `j==i`.
- B. It is false that `j==i`.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

Answer: D

22. Program

_Given:

```

10. public class MyClass {
11.
12. public Integer startingI;
13. public void methodA() {
14. Integer i = new Integer(25);
15. startingI = i;
16. methodB(i);
17. }
18. private void methodB(Integer i2) {
19. i2 = i2.intValue();
20.
21. }
22. }
```

If `methodA` is invoked, which two are true at line 20? (Choose two.)

- A. `i2 == startingI` returns true.
- B. `i2 == startingI` returns false.
- C. `i2.equals(startingI)` returns true.
- D. `i2.equals(startingI)` returns false.

Answer: BC

23. Program

_Given

```

1. public class BuildStuff{
2. public static void main (String
args[]) {
3. Boolean test=new Boolean(true);
4. Integer x=343;
5.Integer y =new BuildStuff().go(test,x);
6. System.out.println(y);
7.}
8. int go(Boolean b,int i){
9. if(b) return (i/7);
10. return (i/49);
```

What is the result?

- A. 7
- B. 49
- C. 343
- D. Compilation fails
- E. An exception is thrown at runtime.

Ans : B

24. Program

Given:

```
11. public class Yikes {
12.
13. public static void go(Long n)
   {System.out.println("Long ");}
14. public static void go(Short n)
   {System.out.println("Short ");}
15. public static void go(int n)
   {System.out.println("int ");}
16. public static void main(String [] args) {
17. short y= 6;
18. long z= 7;
19. go(y);
20. go(z);
21. }
22. }
```

What is the result?

- A. int Long
- B. Short Long
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: A

25. Program

Given:

```
12. public class Wow {
```

```
13. public static void go(short n)
   {System.out.println("short"); }
14. public static void go(Short n)
   {System.out.println("SHORT");}
15. public static void go(Long n)
   {System.out.println(" LONG"); }
16. public static void main(String [] args) {
17. Short y= 6;
18. int z=7;
19. go(y);
20. go(z);
21. }
22. }
```

What is the result?

- A. short LONG
- B. SHORT LONG
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: C

26. Program

Click the Exhibit button.

```
1. public class A {
2. public String doit(int x, int y) {
3. return "a";
4. }
5.
6. public String doit(int... vals) {
7. return "b";
8. }
```

9. }

Given:

25. *A a=new A();*

26. *System.out.println(a.doit(4, 5));*

What is the result?

A. Line 26 prints “a” to System.out.

B. Line 26 prints ‘b’ to System.out.

C. An exception is thrown at line 26 at runtime.

D. Compilation of class A will fail due to an error in line 6.

Answer: A

27. Program

A programmer needs to create a logging method that can accept an arbitrary number of arguments. For example, it may be called in these ways:

`logIt("log message 1 ");`

`logIt("log message2", "log message3");`

`logIt("log message4", "log message5", "log message6");`

Which declaration satisfies this requirement?

A. `public void logIt(String * msgs)`

B. `public void logIt(String [] msgs)`

C. `public void logIt(String... msgs)`

D. `public void logIt(String msg1, String msg2, String msg3)`

Answer: C

28. Program

Given:

10. `public class Bar {`

11. `static void foo(int...x) {`

12. `// insert code here`

13. `}`

14. `}`

Which two code fragments, inserted independently at line 12, will allow the class to compile? (Choose two.)

A. `foreach(x) System.out.println(z);`

B. `for(int z : x) System.out.println(z);`

C. `while(x.hasNext()) System.out.println(x.next());`

D. `for(int i=0; i< x.length; i++)`

`System.out.println(x[i]);`

Answer: BD

29. Program

_Given

12. `public class Barn{`

13. `public static void main (String args[]){`

14. `new Barn().go("hi",1);`

```

15. new Barn().go("hi","world",2);
16.}
17. public void go(String...y,int x){
18.System.out.println(y[y.length-1]+"");
19.}
20.}

```

What is the result?

- A. hi hi
- B. hi world
- C. world world
- D. Compilation fails
- E. An exception is thrown at runtime.

Ans :D

30. Program

Given

```

11. class Mud{
12. //insert code here
13. System.out.println("hi");
14.}
15.}

```

And the following five fragments

- A. public static void main(String...a)
- B. public static void main(String.*a)
- C. public static void main(String... a)
- D. public static void main(String[]...a)
- E. public static void main(String...[]a)

How many of the code fragments, inserted independently at line 12, compile?

- A. 0
- B. 1

- C. 2
- D. 3
- E. 4
- F. 5

Ans: D

31. Program

Given:

```

12. String csv = "Sue,5,true,3";
13. Scanner scanner = new Scanner( csv);
14. scanner.useDelimiter(",");
15. int age = scanner.nextInt();

```

What is the result?

- A. Compilation fails.
- B. After line 15, the value of age is 5.
- C. After line 15, the value of age is 3.
- D. An exception is thrown at runtime.

Answer: D

32. Program

__Given the command line java Pass2 and:

```

15. public class Pass2 {
16. public void main(String [] args) {
17.int x=6;
18. Pass2 p = new Pass2();
19. p.doStuff(x);
20. System.out.print(" main x = "+ x);
21. }
22.

```

```

23. void doStuff(int x) {
24. System.out.print(" doStuffx = "+ x++);
25. }
26. }

```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. doStuffx = 6 main x = 6

D. doStuffx = 6 main x = 7

E. doStuffx = 7 main x = 6

F. doStuffx = 7 main x = 7

Answer: B

33. Program

_Given:

```

15. public class Yippee {
16. public static void main(String [] args) {
17. for(int x = 1; x < args.length; x++) {
18. System.out.print(args[x] + " ");
19. }
20. }
21. }

```

and two separate command line invocations:

java Yippee

java Yippee 1 2 3 4

What is the result?

A. No output is produced.

123

B. No output is produced.

234

C. No output is produced.

1234

D. An exception is thrown at runtime.

123

E. An exception is thrown at runtime.

234

F. An exception is thrown at runtime.

1234

Answer: B

34. Program

_Given:

```

12. public class Yippee2 {
13.
14. static public void main(String [] yahoo)
    {
15. for(int x= 1; x<yahoo.length; x++) {
16. System.out.print(yahoo[x] + " ");
17. }
18. }
19. }

```

and the command line invocation:

java Yippee2 a b c

What is the result?

A. a b

B. b c

C. a b c

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: B

35. Program

_Given:

```

11. public class Commander {
12. public static void main(String[] args) {
13. String myProp = /* insert code here */
14. System.out.println(myProp);
15. }
16. }

```

and the command line:

```

java -Dprop.custom=gobstopper
Commander

```

Which two, placed on line 13, will produce the output gobstopper?

(Choose two.)

- A. `System.load("prop.custom");`
- B. `System.getenv("prop.custom");`
- C. `System.property("prop.custom");`
- D. `System.getProperty("prop.custom");`
- E. `System.getProperties().getProperty("prop.custom");`

Answer: DE

36. Program

_Given:

```

10. public class ClassA {
11. public void count(int i) {
12. count(++i);
13. }

```

14. }

And:

20. `ClassA a = new ClassA();`

21. `a.count(3);`

Which exception or error should be thrown by the virtual machine?

- A. `StackOverflowError`
- B. `NullPointerException`
- C. `NumberFormatException`
- D. `IllegalArgumentException`
- E. `ExceptionInInitializerError`

Answer: A

37. Program

_Given:

```

1. public class Boxer1 {
2. Integer i;
3. int x;
4. public Boxer1(int y) {
5. x=i+y;
6. System.out.println(x);
7. }
8. public static void main(String[] args) {
9. new Boxer1(new Integer(4));
10. }
11. }

```

What is the result?

- A. The value "4" is printed at the command line.

B. Compilation fails because of an error in line 5.

C. Compilation fails because of an error in line 9.

D. A `NullPointerException` occurs at runtime.

E. A `NumberFormatException` occurs at runtime.

F. An `IllegalStateException` occurs at runtime.

Answer: D

38. Program

Given:

```
31. // some code here
32. try {
33. // some code here
34. } catch (SomeException se) {
35. // some code here
36. } finally {
37. // some code here
38. }
```

Under which three circumstances will the code on line 37 be executed?

(Choose three.)

- A. The instance gets garbage collected.
- B. The code on line 33 throws an exception.
- C. The code on line 35 throws an exception.
- D. The code on line 31 throws an exception.
- E. The code on line 33 executes successfully.

Answer: BCE

39. Program

Given:

```
11. public static void parse(String str) {
12. try {
13. float f= Float.parseFloat(str);
14. } catch (NumberFormatException nfe) {
15. f= 0;
16. } finally {
17. System.out.println(f);
18. }
19. }
20. public static void main(String[] args) {
21. parse("invalid");
22. }
```

What is the result?

- A. 0.0
- B. Compilation fails.
- C. A `ParseException` is thrown by the `parse` method at runtime.
- D. A `NumberFormatException` is thrown by the `parse` method at runtime.

Answer: B

40. Program

_Given:

```
33. try {
34. // some code here
35. } catch (NullPointerException e1) {
```

```

36. System.out.print("a");
37. } catch (RuntimeException e2) {
38. System.out.print("b");
39. } finally {
40. System.out.print("c");
41. }

```

What is the result if a `NullPointerException` occurs on line 34?

- A. c
- B. a
- C. ab
- D. ac
- E. bc
- F. abc

Answer: D

41. Program

_Given:

```

11.classA {
12. public void process() {
System.out.print("A,"); } }
13. class B extends A {
14. public void process() throws
IOException {
15. super.process();
16. System.out.print("B,");
17. throw new IOException();
18. } }
19. public static void main(String[] args) {
20. try { new B().process(); }

```

```

21. catch (IOException e) {
System.out.println("Exception"); } }

```

What is the result?

- A. Exception
- B. A,B,Exception
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 14.
- E. A `NullPointerException` is thrown at runtime.

Answer: D

42. Program

_Given:

```

11.classA {
12. public void process() {
System.out.print("A "); } }
13. class B extends A {
14. public void process() throws
RuntimeException {
15. super.process();
16. if (true) throw new RuntimeException();
17. System.out.print("B"); } }
18. public static void main(String[] args) {
19. try { ((A)new B()).process(); }
20. catch (Exception e) {
System.out.print("Exception "); }
21. }

```

What is the result?

- A. Exception
- B. A Exception
- C. A Exception B
- D. A B Exception
- E. Compilation fails because of an error in line 14.
- F. Compilation fails because of an error in line 19.

Answer: B

43. Program

_Given:

```
11. static classA {
12. void process() throws Exception { throw
new Exception(); }
13. }
14. static class B extends A {
15. void process() { System.out.println("B
"); }
16. }
17. public static void main(String[] args) {
18.A a=new B();
19. a.process();
20.}
```

What is the result?

- A. B
- B. The code runs with no output.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 15.

- E. Compilation fails because of an error in line 18.
- F. Compilation fails because of an error in line 19.

Answer: F

44. Program

_Given:

```
11. static class A {
12. void process() throws Exception { throw
new Exception(); }
13. }
14. static class B extends A {
15. void process() {
System.out.println("B"); }
16. }
17. public static void main(String[] args) {
18. new B().process();
19. }
```

What is the result?

- A. B
- B. The code runs with no output.
- C. Compilation fails because of an error in line 12.
- D. Compilation fails because of an error in line 15.
- E. Compilation fails because of an error in line 18.

Answer: A

45. Program

_Given:

```

84. try {
85.   ResourceConnection con =
resourceFactory.getConnection();
86.   Results r = con.query("GET INFO
FROM CUSTOMER");
87.   info = r.getData();
88.   con.close();
89. } catch (ResourceException re) {
90.   errorLog.write(re.getMessage());
91. }
92. return info;

```

Which is true if a ResourceException is thrown on line 86?

- A. Line 92 will not execute.
- B. The connection will not be retrieved in line 85.
- C. The resource connection will not be closed on line 88.
- D. The enclosing method will throw an exception to its caller.

Answer: C

46. Program

_Click the Exhibit button.

```

1. public class A {
2.   public void method1() {
3.     B b=new B();

```

```

4.     b.method2();
5.     // more code here
6.   }
7. }

1. public class B {
2.   public void method2() {
3.     C c=new C();
4.     c.method3();
5.     // more code here
6.   }
7. }

1. public class C {
2.   public void method3() {
3.     // more code here
4.   }
5. }

```

Given:

```

25. try {
26.   A a=new A();
27.   a.method1();
28. } catch (Exception e) {
29.   System.out.print("an error occurred");
30. }

```

Which two are true if a NullPointerException is thrown on line 3 of class C? (Choose two.)

- A. The application will crash.
- B. The code on line 29 will be executed.
- C. The code on line 5 of class A will execute.
- D. The code on line 5 of class B will execute.

E. The exception will be propagated back to line 27.

Answer: BE

47. **Program**

_Click the Exhibit button.

```
1. public class A {
2. public void method1() {
3. try {
4. B b=new B();
5. b.method2();
6. // more code here
7. } catch (TestException te) {
8. throw new RuntimeException(te);
9. }
6. }
7. }
```

```
1. public class B {
2. public void method2() throws
TestException {
3. // more code here
4. }
5. }
```

```
1. public class TestException extends
Exception {
```

```
2. }
```

Given:

```
31. public void method() {
32. A a=new A();
33. a.method1();
34. }
```

Which is true if a TestException is thrown on line 3 of class B?

A. Line 33 must be called within a try block.

B. The exception thrown by method1 in class A is not required to be caught.

C. The method declared on line 31 must be declared to throw a RuntimeException.

D. On line 5 of class A, the call to method2 of class B does not need to be placed in a try/catch block.

Answer: B

48. **Program**

_Given:

```
11. public static void main(String[] args) {
```

```
12. try {
```

```
13. args=null;
```

```
14. args[0] = "test";
```

```
15. System.out.println(args[0]);
```

```
16. } catch (Exception ex) {
```

```
17. System.out.println("Exception");
```

```
18. } catch (NullPointerException npe) {
```

```
19.
```

```
System.out.println("NullPointerException")
```

```
;
```

```
20. }
```

```
21. }
```

What is the result?

A. *test*

B. *Exception*

C. *Compilation fails.*

D. *NullPointerException*

Answer: C

49. Program

_Click the Exhibit button.

SomeException:

```
1. public class SomeException {
2. }
```

Class A:

```
1. public class A {
2. public void doSomething() { }
3. }
```

Class B:

```
1. public class B extends A {
2. public void doSomething() throws
SomeException { }
3. }
```

Which is true about the two classes?

- A. Compilation of both classes will fail.
- B. Compilation of both classes will succeed.
- C. Compilation of class A will fail.
Compilation of class B will succeed.
- D. Compilation of class B will fail.
Compilation of class A will succeed.

Answer: D

50. Program

_Class TestException

```
1. public class TestException extends
Exception {
2. }
```

Class A:

```
1. public class A {
2.
3. public String sayHello(String name)
throws TestException {
4.
5. if(name == null) {
6. throw new TestException();
7. }
8.
9. return "Hello "+ name;
10. }
11.
12. }
```

A programmer wants to use this code in an application:

45. *A a=new A();*

46. *System.out.println(a.sayHello("John"));*

Which two are true? (Choose two.)

- A. *Class A will not compile.*
- B. *Line 46 can throw the unchecked exception TestException.*
- C. *Line 45 can throw the unchecked exception TestException.*

D. Line 46 will compile if the enclosing method throws a `TestException`.

E. Line 46 will compile if enclosed in a try block, where `TestException` is caught.

Answer: DE

51. Program

Given:

```
11. static void test() {
12. try {
13. String x=null;
14. System.out.print(x.toString() +“ “);
15. }
16. finally { System.out.print(“finally “); }
17. }
18. public static void main(String[] args) {
19. try { test(); }
20. catch (Exception ex) {
    System.out.print(“exception “); }
21. }
```

What is the result?

- A. null*
- B. finally*
- C. null finally*
- D. Compilation fails.*
- E. finally exception*

Answer: E

52. Program

Given:

```
10. public class Foo {
11. static int[] a;
12. static { a[0]=2; }
13. public static void main( String[] args) {}
14. }
```

Which exception or error will be thrown when a programmer attempts to run this code?

- A. `java.lang. StackOverflowError`
- B. `java.lang. IllegalStateException`
- C. `java.lang. ExceptionInInitializerError`
- D. `java.lang. ArrayIndexOutOfBoundsException`

Answer: C

53. Program

Given:

```
11. static void test() throws
    RuntimeException {
12. try {
13. System.out.print(“test “);
14. throw new RuntimeException();
15. }
16. catch (Exception ex) {
    System.out.print(“exception “); }
```

```

17. }
18. public static void main(String[] args) {
19. try { test(); }
20. catch (RuntimeException ex) {
    System.out.print("runtime "); }
21. System.out.print("end ");
22. }

```

What is the result?

- A. test end
- B. Compilation fails.
- C. test runtime end
- D. test exception end
- E. A Throwable is thrown by main at runtime.

Answer: D

54. Program

```

5. class A{
    6. void foo() throws
      Exception{throw new Exception();}
    7.}
    8. class SubB2 extends A{
    9.void
      foo(){System.out.println("B");}
    10.}
    11. class Tester{
    12. public static void main(String
      args[]){
    13. A a=new SubB2();
    14. a.foo();
    15.}

```

```

16.}

```

What is the result?

- A. B
- B. B, followed by Exception
- C. Compilation fails due to an error on line 9
- D. Compilation fails due to an error on line 14
- E. An exception is thrown with no other output.

Ans : D

55. Program

_Given:

```

8. public class test {
9. public static void main(String [] a) {
10. assert a.length == 1;
11. }
12. }

```

Which two will produce an AssertionError?

(Choose two.)

- A. java test
- B. java -ea test
- C. java test file1
- D. java -ea test file1
- E. java -ea test file1 file2
- F. java -ea:test test file1

Answer: BE

56. Program

__Given:

```

12. public class AssertStuff {
13.
14. public static void main(String [] args) {
16. int y= 7;
17. int x=5;
18. assert (x> y): "stuff";
19. System.out.println("passed");
20. }
21. }

```

And these command line invocations:

```

java AssertStuff
java -ea AssertStuff

```

What is the result?

- A. passed
stuff
- B. stuff
passed
- C. passed

An AssertionError is thrown with the word "stuff" added to the stack trace.

- D. passed

An AssertionError is thrown without the word "stuff" added to the stack trace.

- E. passed

An AssertionError is thrown with the word "stuff" added to the stack trace.

- F. passed

An AssertionError is thrown without the word "stuff" added to the stack trace.

Answer: C

57. Program

__Click the Exhibit button.

```

1. public class Test {
2.
3. public static void main(String [] args) {
4. boolean assert = true;
5. if(assert) {
6. System.out.println("assert is true");
7. }
8. }
9.
10. }

```

Given:

javac -source 1.3 Test.java

What is the result?

- A. *Compilation fails.*
- B. *Compilation succeeds with errors.*
- C. *Compilation succeeds with warnings.*
- D. *Compilation succeeds without warnings or errors.*

Answer: C

58. Program

_Given:

23. *int z=5;*

24.

25. *public void stuff1(int x) {*

26. *assert (x > 0);*

27. *switch(x) {*

28. *case 2: x = 3;*

29. *default: assert false; } }*

30.

31. *private void stuff2(int y) { assert (y < 0); }*

32.

33. *private void stuff3() { assert (stuff4O); }*

34.

35. *private boolean stuff4() { z = 6; return false; }*

Which is true?

A. All of the assert statements are used appropriately.

B. Only the assert statement on line 31 is used appropriately.

C. The assert statements on lines 29 and 31 are used appropriately.

D. The assert statements on lines 26 and 29 are used appropriately.

E. The assert statements on lines 29 and 33 are used appropriately.

F. The assert statements on lines 29, 31, and 33 are used appropriately.

G. The assert statements on lines 26, 29, and 31 are used appropriately.

Answer: C

59. Program

_Given:

11. *static void test() throws Error {*

12. *if (true) throw new AssertionError();*

13. *System.out.print("test ");*

14. *}*

15. *public static void main(String[] args) {*

16. *try { test(); }*

17. *catch (Exception ex) {*

System.out.print("exception "); }

18. *System.out.print("end ");*

19. *}*

What is the result?

A. end

B. Compilation fails.

C. exception end

D. exception test end

E. A Throwable is thrown by main.

F. An Exception is thrown by main.

Answer: E

60. Program

__ Given a method that must ensure that its parameter is not null:

```
11. public void someMethod(Object value) {
12. // check for null value
....
20. System.out.println(value.getClass());
21. }
```

What, inserted at line 12, is the appropriate way to handle a null value?

- A. assert value == null;
- B. assert value != null, "value is null";
- C. if (value == null) {
throw new AssertionError("value is null");
- D. if (value == null) {
throw new
IllegalArgumentException("value is null");

Answer: D

61. Program

__ Which two classes correctly implement both the java.lang.Runnable and the java.lang.Cloneable interfaces? (Choose two.)

- A. public class Session
implements Runnable, Cloneable {
public void run();
public Object clone();
}
- B. public class Session
extends Runnable, Cloneable {
public void run() { / do something */ }
public Object clone() { / make a copy */ }
}
- C. public class Session
implements Runnable, Cloneable {
public void run() { / do something */ }
public Object clone() { /* make a copy */ }
}
- D. public abstract class Session
implements Runnable, Cloneable {
public void run() { / do something */ }
public Object clone() { /*make a copy */ }
}
- E. public class Session
implements Runnable, implements Cloneable
{
public void run() { / do something */ }
public Object clone() { / make a copy */ }
}

Answer: CD

62. Program

```

11. public class PingPong implements
Runnable{
12. synchronized void hit(long n){
13. for(int i=1;i<3;i++)
14. System.out.println(n+"-"+i+"");
15.}
16. public static void main(String args[]){
17. new Thread(new PingPong()).start();
18. new Thread(new PingPong()).start();
19.}
20. public void run(){
21. hit(Thread.currentThread().getId());
22.}
23.}

```

Which two statements are true?(choose two)

- A. The output could be 8-1 7-2 8-2 7-1
- B. The output could be 7-1 7-2 8-1 6-1
- C. The output could be 8-1 7-1 7-2 8-2
- D. The output could be 8-1 8-2 7-1 7-2

Ans : C, D

63. Program

Given

```

11. public class PingPong2 {
12. synchronized void hit(long n){
13. for(int i=1;i<3;i++)
14. System.out.println(n+"-"+i+"");
15.}}
16. public class Tester implements
Runnable{
17. static PingPong2 pp2=new PingPong2();

```

```

18. public static void main(String args[]){
19. new Thread(new Tester()).start();
20. new Thread(new Tester()).start();
21.}
22. public void run(){
23. pp2.hit(Thread.currentThread().getId());
24.}
25.}

```

Which two statements are true?(choose two)

- A. The output could be 5-1 6-1 6-2 5-2
- B. The output could be 6-1 6-2 5-1 5-2
- C. The output could be 6-1 5-2 6-2 5-1
- D. The output could be 6-1 6-2 5-1 7-1

Ans: B

64. Program

_Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)

- A. new Thread() {
public void run() { doStuff(); }
}
- B. new Thread() {
public void start() { doStuff(); }
}
- C. new Thread() {
public void start() { doStuff(); }
} .run();

D. `new Thread() {
public void run() { doStuff(); }
} .start();`

E. `new Thread(new Runnable() {
public void run() { doStuff(); }
}).run();`

F. `new Thread(new Runnable() {
public void run() { doStuff(); }
}).start();`

Answer: DF

65. Program

Given:

```
1. public class Threads3 implements
   Runnable {
2. public void run() {
3. System.out.print("running");
4. }
5. public static void main(String[] args) {
6. Thread t = new Thread(new Threads3());
7. t.run();
8. t.run();
9. t.start();
10. }
11. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "running".

D. The code executes and prints
"runningrunning".

E. The code executes and prints
"runningrunningrunning".

Answer: E

66. Program

Given:

```
1. public class Threads4 {
2. public static void main (String[] args) {
3. new Threads4().go();
4. }
5. public void go() {
6. Runnable r = new Runnable() {
7. public void run() {
8. System.out.print("foo");
9. }
10. };
11. Thread t = new Thread(r);
12. t.start();
13. t.start();
14. }
15. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints
"foo".

D. The code executes normally, but nothing is printed.

Answer: B

67. **Program**

Given:

```
1. public class Threads5 {
2. public static void main (String[] args) {
3. new Thread(new Runnable() {
4. public void run() {
5. System.out.print("bar");
6. }).start();
7. }
8. }
```

What is the result?

- A. Compilation fails.*
- B. An exception is thrown at runtime.*
- C. The code executes normally and prints "bar".*
- D. The code executes normally, but nothing prints.*

Answer: C

68. **Program**

Given:

```
11. Runnable r = new Runnable() {
12. public void run() {
13. System.out.print("Cat");
```

```
14. }
```

```
15. };
```

```
16. Thread t = new Thread(r) {
```

```
17. public void run() {
```

```
18. System.out.print("Dog");
```

```
19. }
```

```
20. };
```

```
21. t.start();
```

What is the result?

- A. Cat*
- B. Dog*
- C. Compilation fails.*
- D. The code runs with no output.*
- E. An exception is thrown at runtime.*

Answer: B

69. **Program**

Click the Exhibit button.

Given:

```
10. public class Starter extends Thread {
11. private int x= 2;
12. public static void main(String[] args)
throws Exception {
13. new Starter().makeItSo();
14. }
15. public Starter() {
16. x=5;
17. start();
18. }
```

```

19. public void makeItSo() throws
Exception {
20. join();
21. x=x- 1;
22. System.out.println(x);
23. }
24. public void run() { x *= 2; }
25. }

```

What is the output if the main() method is run?

- A. 4
- B. 5
- C. 8
- D. 9
- E. Compilation fails.
- F. An exception is thrown at runtime.
- G. It is impossible to determine for certain.

Answer: D

70. Program

Given:

```

1. public class Threads2 implements
Runnable {
2.
3. public void run() {
4. System.out.println("run.");
5. throw new
RuntimeException("Problem");
6. }
7. public static void main(String[] args) {
8. Thread t = new Thread(new Threads2());

```

```

9. t.start();
10. System.out.println("End of method.");
11. }
12. }

```

Which two can be results? (Choose two.)

- A. java.lang.RuntimeException: Problem
- B. run.
- java.lang.RuntimeException: Problem
- C. End of method.
- java.lang.RuntimeException: Problem
- D. End of method.
- run.
- java.lang.RuntimeException: Problem
- E. run.
- java.lang.RuntimeException: Problem
- End of method.

Answer: DE

71. Program

Given:

```

1. public class TestOne {
2. public static void main (String[] args)
throws Exception {
3. Thread.sleep(3000);
4. System.out.println("sleep");
5. }
6. }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.

- C. The code executes normally and prints “sleep”.
- D. The code executes normally, but nothing is printed.

Answer: C

72. Program

_Given:

```
1. public class TestOne implements
   Runnable {
2. public static void main (String[] args)
   throws Exception {
3. Thread t = new Thread(new TestOne());
4. t.start();
5. System.out.print("Started");
6. t.join();
7. System.out.print("Complete");
8. }
9. public void run() {
10. for (int i= 0; i< 4; i++) {
11. System.out.print(i);
12. }
13. }
14. }
```

What can be a result?

- A. Compilation fails.
- B. An exception is thrown at runtime.

C. The code executes and prints “StartedComplete”.

D. The code executes and prints “StartedComplete0123”.

E. The code executes and prints “Started0123Complete”.

Answer: E

73. Program

_Which two statements are true? (Choose two)

- A. It is possible for more than two threads to deadlock at once.
- B. the JVM implementation guarantees that multiple threads can not enter into a deadlock state.
- C. Deadlocked threads release once their sleep() methods sleep duration has expired.
- D. Deadlocking can occur only when the wait(),notify(),and notifyAll() methods are used incorrectly.
- E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.
- F. If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting invocation of

Thread.yield();

Ans: A, F

74. **Program**

_Click the Exhibit button.

Given:

```

1. public class TwoThreads {
2
3. private static Object resource = new
Object();
4.
5. private static void delay(long n) {
6. try { Thread.sleep(n); }
7. catch (Exception e) {
System.out.print("Error "); }
8. }
9
10. public static void main(String[] args) {
11. System.out.print("StartMain ");
12. new Thread1().start();
13. delay(1000);
14. Thread t2 = new Thread2();
15. t2.start();
16. delay(1000);
17. t2.interrupt
18. delay(1000);
19. System.out.print("EndMain ");
20. }
21.
22. static class Thread 1 extends Thread {

```

```

23. public void run() {
24. synchronized (resource) {
25. System.out.print("Start1 ");
26. delay(6000);
27. System.out.print("End1 ");
28. }
29. }
30. }
31.
32. static class Thread2 extends Thread {
33. public void run() {
34. synchronized (resource) {
35. System.out.print("Start2 ");
36. delay(2000);
37. System.out.print("End2 ");
38. }
39. }
40. }
41. }

```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time. What is the output if the main() method is run?

A. Compilation fails.
 B. Deadlock occurs.
 C. StartMain Start1 Error EndMain End1
 D. StartMain Start1 EndMain End1 Start2 End2
 E. StartMain Start1 Error Start2 EndMain End2 End1

F. StartMain Start1 Start2 Error End2
EndMain End1

G. StartMain Start1 EndMain End1 Start2
Error End2

Answer: G

D. declare the constructor using the
synchronized keyword

E. declare increment() using the
synchronized keyword

Answer: ACE

75. Program

_Given:

```
public class NamedCounter {
    private final String name;
    private int count;
    public NamedCounter(String name) {
        this.name = name; }
    public String getName() { return name; }
    public void increment() { count++; }
    public int getCount() { return count; }
    public void reset() { count = 0; }
}
```

Which three changes should be made to
adapt this class to be used
safely by multiple threads? (Choose three.)

- A. declare reset() using the synchronized keyword
- B. declare getName() using the synchronized keyword
- C. declare getCount() using the synchronized keyword

76. Program

_Click the Exhibit button:

```
1. public class Threads 1 {
2.     intx=0;
3.     public class Runner implements Runnable
4.     {
5.         int current = 0;
6.         for(int i=0;i<4;i++){
7.             current = x;
8.             System.out.print(current + " ");
9.             x = current + 2;
10.        }
11.    }
12. }
13.
14. public static void main(String[] args) {
15.     new Threads1().go();
16. }
17.
18. public void go() {
```


19. *Runnable r1 = new Runner();*

20. *new Thread(r1).start();*

21. *new Thread(r1).start();*

22. *}*

23. *}*

Which two are possible results? (Choose two.)

A. 0, 2, 4, 4, 6, 8, 10, 12,

B. 0, 2, 4, 6, 8, 10, 2, 4,

C. 0, 2, 4, 6, 8, 10, 12, 14,

D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,

E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

Answer: AC

77. **Program**

_Click the Exhibit button.

1. *import java.util.*;*

2.

3. *public class NameList {*

4. *private List names = new ArrayList();*

5. *public synchronized void add(String name) { names.add(name); }*

6. *public synchronized void printAll() {*

7. *for (int i = 0; i < names.size(); i++) {*

8. *System.out.print(names.get(i) + " ");*

9. *}*

10. *}*

11. *public static void main(String[] args) {*

12. *final NameList sl = new NameList();*

13. *for(int i=0;i<2;i++) {*

14. *new Thread() {*

15. *public void ruin() {*

16. *sl.add("A");*

17. *sl.add("B");*

18. *sl.add("C");*

19. *sl.printAll();*

20. *}*

21. *}.start();*

22. *}*

23. *}*

24. *}*

Which two statements are true if this class is compiled and run?

(Choose two.)

A. An exception may be thrown at runtime.

B. The code may run with no output, without exiting.

C. The code may run with no output, exiting normally.

D. The code may run with output "A B A B C C", then exit.

E. The code may run with output "A B C A B C A B C", then exit.

F. The code may run with output "A A A B C A B C C", then exit.

G. The code may run with output "A B C A A B C A B C", then exit.

Answer: EG

78. Program

_Given:

```

1. public class TestFive {
2. private int x;
3. public void foo() {
4 int current = x;
5. x = current + 1;
6. }
7. public void go() {
8. for(int i=0;i<5;i++) {
9. new Thread() {
10. public void run() {
11. foo();
12. System.out.print(x + " ");
13. } }.start();
14. }}}

```

Which two changes, taken together, would guarantee the output: 1, 2,

3, 4, 5, ? (Choose two.)

- A. Move the line 12 print statement into the foo() method.
- B. Change line 7 to public synchronized void go() {.
- C. Change the variable declaration on line 3 to private volatile int x;.
- D. Wrap the code inside the foo() method with a synchronized(this) block.
- E. Wrap the for loop code inside the go() method with a synchronized

block synchronized(this) { // for loop code here }.

Answer: AD

79. Program

_Which three will compile and run without exception? (Choose three.)

- A. private synchronized Object o;
- B. void go() {
 synchronized() { /* code here */ }
}
- C. public synchronized void go() { /* code here */ }
- D. private synchronized(this) void go() { /* code here */ }
- E. void go() {
 synchronized(Object.class) { /* code here */ }
}
- F. void go() {
 Object o = new Object();
 synchronized(o) { /* code here */ }
}

Answer: CEF

80. Program

_Given:

```

1. public class TestSeven extends Thread {
2. private static int x;

```

```

3. public synchronized void doThings() {
4. int current = x;
5. current++;
6. x = current;
7. }
8. public void run() {
9. doThings();
10. }
11. }

```

Which is true?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. Synchronizing the run() method would make the class thread-safe.
- D. The data in variable "x" are protected from concurrent access problems.
- E. Declaring the doThings() method as static would make the class thread-safe.
- F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

Answer: E

81. Program

_Click the Exhibit button.

```

10. public class Transfers {

```

```

11. public static void main(String[] args)
    throws Exception {
12. Record r1 = new Record();
13. Record r2 = new Record();
14. doTransfer(r1, r2, 5);
15. doTransfer(r2, r1, 2);
16. doTransfer(r1, r2, 1);
17. // print the result
18. System.out.println("r1 = " + r1.get() + ",
    r2=" + r2.get());
19. }
20. private static void doTransfer(
21. final Record a, final Record b, final int
    amount) {
22. Thread t = new Thread() {
23. public void run() {
24. new Clerk().transfer(a, b, amount);
25. }
26. };
27. t.start();
28. }
29. }
30. class Clerk {
31. public synchronized void
    transfer(Record a, Record b, int amount){
32. synchronized (a) {
33. synchronized (b) {
34. a.add(-amount);
35. b.add(amount);
36. }
37. }

```

```

38. }
39. }
40. class Record {
41. int num=10;
42. public int get() { return num; }
43. public void add(int n) { num = num + n;
}
44. }

```

If *Transfers.main()* is run, which three are true? (Choose three.)

- A. The output may be “r1 = 6, r2 = 14”.
- B. The output may be “r1 = 5, r2 = 15”.
- C. The output may be “r1 = 8, r2 = 12”.
- D. The code may run (and complete) with no output.
- E. The code may deadlock (without completing) with no output.
- F. *M IllegalStateException* or *InterruptedException* may be thrown at runtime.

Answer: ABE

82. Program

Click the Exhibit button.

```

1. class Computation extends Thread {
2.
3. private int num;
4. private boolean isComplete;
5. private int result;
6.

```

```

7. public Computation(int num) { this.num =
num; }
8.
9. public synchronized void run() {
10. result = num * 2;
11. isComplete = true;
12. notify();
13. }
14.
15. public synchronized int getResult() {
16. while (!isComplete) {
17. try {
18. wait();
19. } catch (InterruptedException e) {}
20. }
21. return result;
22. }
23.
24. public static void main(String[] args) {
25. Computation[] computations = new
Computation [4];
26. for (int i = 0; i < computations.length;
i++) {
27. computations[i] = new Computation(i);
28. computations[i].start();
29. }
30. for (Computation c : computations)
31. System.out.print(c.getResult() + “ ”);
32. }
33. }

```

What is the result?

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".
- E. The code may run with output "2 0 6 4".
- F. The code may run with output "0 2 4 6".

Answer: F

83. Program

_Given:

```
7. void waitForSignal() {
8. Object obj = new Object();
9. synchronized (Thread.currentThread()) {
10. obj.wait();
11. obj.notify();
12. }
13. }
```

Which is true?

- A. This code may throw an InterruptedException.
- B. This code may throw an IllegalStateException.
- C. This code may throw a TimeoutException after ten minutes.
- D. This code will not compile unless "obj.wait()" is replaced with

"((Thread) obj).wait()".

- E. Reversing the order of obj.wait() and obj.notify() may cause this method to complete normally.
- F. A call to notify() or notifyAll() from another thread may cause this method to complete normally.

Answer: B

84. Program

_Given:

foo and bar are public references available to many other threads. foo refers to a Thread and bar is an Object. The thread foo is currently executing bar.wait(). From another thread, which statement is the most reliable way to ensure that foo will stop executing wait()?

- A. foo.notify();
- B. bar.notify();
- C. foo.notifyAll();
- D. Thread.notify();
- E. bar.notifyAll();
- F. Object.notify();

Answer: E

85. Program

_Given that t1 is a reference to a live thread, which is true?

- A. The Thread.sleep() method can take t1 as an argument.
- B. The Object.notify() method can take t1 as an argument.
- C. The Thread.yield() method can take t1 as an argument.
- D. The Thread.setPriority() method can take t1 as an argument.
- E. The Object.notify() method arbitrary chooses which thread to notify.

Ans: E

86. Program

_Given that Traingle implement Runnable,and;

```

31.void go()throws Exception{
32.Thread t=new Thread(new Triangle());
33.t.start();
34.for(int x=1;x<100000;x++){
35.//insert code here
36.if(x%100==0)System.out.println("g");
37.}}
38.public void run(){
39.try{
40.for(int x=1;x<100000;x++)
41.//insert the same code here

```

```

42.if(x%100==0)System.out.println("t");
43.}
44.}catch(Exception e){}
45.}

```

Which two statements,insert independently at both lines

35 and 41,tend allow both threas to temporarily pause and allow the other threads to execute”
(Choose two)

- A.Thread.wait()
- B.Thread.join()
- C.Thread.yield()
- D.Thread.sleep(1)
- E.Thread.notify()

Answer:C,D

87. Program

_Given:

```

42. public class ClassA {
43. public int getValue() {
44.int value=0;
45. boolean setting = true;
46. String title="Hello";
47. if (value || (setting && title ==
"Hello")) { return 1; }
48. if (value == 1 & title.equals("Hello")) {
return 2; }
49. }
50. }

```

And:

70. *ClassA a = new ClassA();*

71. *a.getValue();*

What is the result?

A. 1

B. 2

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

Answer: C

88. Program

_Given:

```
11. public class Person {
12.     private String name, comment;
13.     private int age;
14.     public Person(String n, int a, String c) {
15.         name = n; age = a; comment = c;
16.     }
17.     public boolean equals(Object o) {
18.         if(! (o instanceof Person)) return false;
19.         Person p = (Person)o;
20.         return age == p.age &&
21.             name.equals(p.name);
22.     }
23. }
```

What is the appropriate definition of the hashCode method in class

Person?

A. *return super.hashCode();*

B. *return name.hashCode() + age * 7;*

C. *return name.hashCode() +*

comment.hashCode() /2;

D. *return name.hashCode() +*

*comment.hashCode() / 2 - age * 3;*

Answer: B

89. program

_Given:

```
11. rbo = new ReallyBigObject();
12. // more code here
13. rbo = null;
14. /* insert code here */
```

Which statement should be placed at line 14 to suggest that the virtual machine expend effort toward recycling the memory used by the object rbo?

A. *System.gc();*

B. *Runtime.gc();*

C. *System.freeMemory();*

D. *Runtime.getRuntime().growHeap();*

E. *Runtime.getRuntime().freeMemory();*

Answer: A

90. Program

Which two are true? (Choose two.)

A. A finalizer may NOT be invoked explicitly.

- B. The finalize method declared in class Object takes no action.
- C. `super.finalize()` is called implicitly by any overriding finalize method.
- D. The finalize method for a given object will be called no more than once by the garbage collector.
- E. The order in which finalize will be called on two objects is based on the order in which the two objects became finalizable.

Answer: BD

91. Program

_Which statements are true?

- A. A class `finalize()` method cannot be invoked explicitly.
- B. `super.finalize()` is called explicitly by any overriding `finalize()` method.
- C. The `finalize()` method for a given Object is called no more than once by the garbage collector.
- D. The order in which `finalize()` is called on two objects is based on the order in which two objects became finalizable.

Ans :C

92. Program

_Given:

11. class Snoochy {

```

12. Boochybooch;
13. public Snoochy() { booch = new
    Boochy(this); }
14. }
15.
16. class Boochy {
17. Snoochy snooch;
18. public Boochy(Snoochy s) { snooch = s;
    }
19. }

```

And the statements:

```

21. public static void main(String[] args) {
22. Snoochy snoog = new Snoochy();
23. snoog = null;
24. // more code here
25. }

```

Which statement is true about the objects referenced by snoog, snooch, and booch immediately after line 23 executes?

- A. None of these objects are eligible for garbage collection.
- B. Only the object referenced by booch is eligible for garbage collection.
- C. Only the object referenced by snoog is eligible for garbage collection.
- D. Only the object referenced by snooch is eligible for garbage collection.

E. The objects referenced by snooch and booch are eligible for garbage collection.

Answer: E

93. Program

Given:

```
1. public class GC {
2.     private Object o;
3.     private void doSomethingElse(Object obj)
       { o = obj; }
4.     public void doSomething() {
5.         Object o = new Object();
6.         doSomethingElse(o);
7.         o = new Object();
8.         doSomethingElse(null);
9.     }
10. }
11. }
```

When the doSomething method is called, after which line does the Object created in line 5 become available for garbage collection?

- A. Line 5
- B. Line 6
- C. Line 7
- D. Line 8
- E. Line 9
- F. Line 10

Answer: D

94. Program

Given:

```
11. public static void test(String str) {
12.     int check = 4;
13.     if (check == str.length()) {
14.         System.out.print(str.charAt(check - 1)
15.             + ", ");
16.     } else {
17.         System.out.print(str.charAt(0) + ", ");
18.     }
19. }
```

and the invocation:

```
21. test("four");
22. test("tee");
23. test("to");
```

What is the result?

- A. r, t, t,
- B. r, e, o,
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: C

95. Program

Given:

```
11. public static void test(String str) {
12.     if(str == null || str.length() == 0) {
13.         System.out.println("String is empty");
14.     } else {
```

15. `System.out.println("String is not empty");`

16. `}`

17. `}`

And the invocation:

31. `test(null);`

What is the result?

A. An exception is thrown at runtime.

B. "String is empty" is printed to output.

C. Compilation fails because of an error in line 12.

D. "String is not empty" is printed to output.

Answer: A

96. Program

Which two scenarios are not safe to replace a `StringBuffer` object with a `StringBuilder` object? (Choose two)

A. When using versions of Java technology earlier than 5.0.

B. When sharing a `StringBuffer` among multiple threads.

C. When using the `java.io.class` `StringBufferInputStream`.

D. When you plan to reuse the `StringBuffer` to build more than one `String`.

Ans: A, B

97. Program

Given

22. `StringBuilder sb1=new`

`StringBuilder("123");`

23. `String s1="123";`

24. //insert code here

25. `System.out.println(sb1+""+s1);`

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

A. `sb1.append("abc");s1.append("abc");`

B. `sb1.append("abc");s1.concat("abc");`

C. `sb1.concat("abc");s1.append("abc");`

D. `sb1.concat("abc");s1.concat("abc");`

E. `sb1.append("abc");s1=s1.concat("abc");`

F. `sb1.concat("abc");s1=s1.concat("abc");`

G. `sb1.append("abc");s1=s1+s1.concat("abc");`

H. `sb1.concat("abc");s1=s1+s1.concat("abc");`

Ans : E

98. Program

Given

1. `public class KungFu{`

2. `public static void main (String args[]){`

3. `Integer x=400;`

4. `Integer y=x;`

5. `x++;`

6. `StringBuilder sb1=new`

`StringBuilder("123");`

7. `StringBuilder sb2=sb1;`

```

8. sb1.append("5");
9.
System.out.println((x==y)+" "+(sb1==sb2));
10.}
11.}

```

What is the result?

- A. true true
- B. false true
- C. true false
- D. false false
- E. Compilation fails
- F. An exception is thrown at runtime

Ans : B

99. Program

_Given:

```

1. public class TestString 1 {
2. public static void main(String[] args)
3. {
4. String str = "420";
5. str += 42;
6. System.out.print(str);
7. }

```

What is the output?

- A. 42
- B. 420
- C. 462
- D. 42042
- E. Compilation fails.
- F. An exception is thrown at runtime.

Answer: D

100. Program

_Given this method in a class:

```

21. public String toString() {
22. StringBuffer buffer = new
StringBuffer();
23. buffer.append('<');
24. buffer.append(this.name);
25. buffer.append('>');
26. return buffer.toString();
27. }

```

Which is true?

- A. This code is NOT thread-safe.
- B. The programmer can replace *StringBuffer* with *StringBuilder* with no other changes.
- C. This code will perform well and converting the code to use *StringBuilder* will not enhance the performance.
- D. This code will perform poorly. For better performance, the code

should be rewritten: return "<"+
this.name + ">";

Answer: B

101. Program

_Given:

```
1. public class MyLogger {
2. private StringBuilder logger = new
   StringBuuilder();
3. public void log(String message, String
   user) {
4. logger.append(message);
5. logger.append(user);
6. }
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system. How must this code be changed to be thread-safe?

- A. synchronize the log method*
- B. replace StringBuilder with StringBuffer*
- C. No change is necessary, the current MyLogger code is already thread-safe.*
- D. replace StringBuilder with just a String object and use the string concatenation (+) within the log method*

Answer: A

102. Program

_Given:

```
11. public String makinStrings() {
12. String s = "Fred";
13. s = s + "47";
```

```
14. s = s.substring(2, 5);
15. s = s.toUpperCase();
16. return s.toString();
17. }
```

How many String objects will be created when this method is invoked?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5
- F. 6

Answer: C

103. Program

_Given:

```
1. public class TestString3 {
2. public static void main(String[] args) {
3. // insert code here
5. System.out.println(s);
6. }
7. }
```

Which two code fragments, inserted independently at line 3, generate the output 4247? (Choose two.)

- A. String s = "123456789";
s = (s-"123").replace(1,3,"24") - "89";
- B. StringBuffer s = new
StringBuffer("123456789");
s.delete(0,3).replace(1,3, "24").delete(4,6);

C. StringBuffer s = new
StringBuffer("123456789");
s.substring(3,6).delete(1 ,3).insert(1, "24");

D. StringBuilder s = new
StringBuilder("123456789");
s.substring(3,6).delete(1 ,2).insert(1, "24");

E. StringBuilder s = new
StringBuilder("123456789");
s.delete(0,3).delete(1 ,3).delete(2,5).insert(
1, "24");

Answer: BE

104. Program

_Given:

11. String test = "This is a test";
12. String[] tokens = test.split("s");
13. System.out.println(tokens.length);

What is the result?

- A. 0
- B. 1
- C. 4
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: D

105. Program

_Given:

11. String test= "a1b2c3";
12. String[] tokens = test.split("\\d");
13. for(String s: tokens) System.out.print(s
+ " ");

What is the result?

- A. a b c
- B. 1 2 3
- C. a1b2c3
- D. a1 b2 c3
- E. Compilation fails.
- F. The code runs with no output.
- G. An exception is thrown at runtime.

Answer: A

106. Program

_Given:

11. String test = "Test A. Test B. Test C.";
12. // insert code here

13. String[] result = test.split(regex);

Which regular expression inserted at line 12
will correctly split test into
"Test A," "Test B," and "Test C"?

- A. String regex = ".";
- B. String regex = " ";
- C. String regex = ".*";
- D. String regex = "\\s";
- E. String regex = "\\s.*";
- F. String regex = "\\w[\\.]+";

Answer: E

107. Program

_Given:

```
12. System.out.format("Pi is approximately
%d.", Math.PI);
```

What is the result?

- A. Compilation fails.
- B. Pi is approximately 3.
- C. Pi is approximately 3.141593.
- D. An exception is thrown at runtime.

Answer: D

108. Program

_Given

```
1. public class LineUp{
2. public static void main(String args[]){
3. double d=12.345;
4. //insert code here
5.}
6.}
```

Which code fragment, inserted at line 4 produces the output |2.345|?

- A. System.out.printf("|%6d\n",d);
- B. System.out.printf("|%6f\n",d);
- C. System.out.printf("|%3.7d\n",d);
- D. System.out.printf("|%3.7f\n",d);
- E. System.out.printf("|%6.3d\n",d);

F. System.out.printf("|%6.3f\n",d);

Ans:F

109. Program

_Given:

- d is a valid, non-null Date object
- df is a valid, non-null DateFormat

object set to the current locale .

What outputs the current locales country name and the appropriate version of d's date?

A. Locale loc = Locale.getLocale();
 System.out.println(loc.getDisplayCountry()
 + " "+ df.format(d));

B. Locale loc = Locale.getDefault();
 System.out.println(loc.getDisplayCountry()
 + " "+ df.format(d));

C. Locale bc = Locale.getLocale();
 System.out.println(loc.getDisplayCountry()
 + " "+ df.setDateFormat(d));

D. Locale loc = Locale.getDefault();
 System.out.println(loc.getDispbayCountry()
 + " "+ df.setDateFormat(d));

Answer: B

110. Program

_Given:

```

12. NumberFormat nf=
    NumberFormat.getInstance();
13. nf.setMaximumFractionDigits(4);
14. nf.setMinimumFractionDigits(2);
15. String a = nf.format(3.1415926);
16. String b = nf.format(2);

```

Which two are true about the result if the default locale is Locale.US?

(Choose two.)

- A. The value of b is 2.
- B. The value of a is 3.14.
- C. The value of b is 2.00.
- D. The value of a is 3.141.
- E. The value of a is 3.1415.
- F. The value of a is 3.1416.
- G. The value of b is 2.0000.

Answer: CF

111. Program

_Given:

```

11. double input = 314159.26;
12. NumberFormat nf=
    NumberFormat.getInstance(Locale.ITALIA
N);
13. String b;
14. //insert code here

```

Which code, inserted at line 14, sets the value of b to 3 14.159,26?

- A. b = nf.parse(input);
- B. b = nf.format(input);
- C. b = nf.equals(input);
- D. b = nf.parseObject(input);

Answer: B

112. Program

_Given:

```

14. DateFormat df;
15. Date date = new Date();
16. //insert code here
17. String s = df.format( date);

```

Which two, inserted independently at line 16, allow the code to compile? (Choose two.)

- A. df= new DateFormat();
- B. df= Date.getFormatter();
- C. df= date.getFormatter();
- D. df= date.getDateFormatter();
- E. df= Date.getDateFormatter();
- F. df= DateFormat.getInstance();
- G. df= DateFormat.getDateInstance();

Answer: FG

113. Program

_Given:

```

12. Date date = new Date();
13. df.setLocale(Locale.ITALY);

```

14. `String s = df.format(date);`

The variable `df` is an object of type `DateFormat` that has been initialized in line 11. What is the result if this code is run on December 14, 2000?

- A. The value of `s` is 14-dic-2004.
- B. The value of `s` is Dec 14, 2000.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

Answer: D

114. Program

_Given:

```
33. Date d = new Date(0);
34. String ds = "December 15, 2004";
35. // insert code here
36. try {
37. d = df.parse(ds);
38. }
39. catch(ParseException e) {
40. System.out.println("Unable to parse "+
ds);
41. }
42. // insert code here too
```

Which will create the appropriate `DateFormat` object and add a day to the `Date` object?

- A. 35. `DateFormat df= DateFormat.getDateFormat();`
42. `d.setTime((60 * 60 * 24) + d.getTime());`
- B. 35. `DateFormat df= DateFormat.getDateInstance();` 42. `d.setTime((1000 * 60 * 60 * 24) + d.getTime());`
- C. 35. `DateFormat df= DateFormat.getDateFormat();`
42. `d.setLocalTime((1000*60*60*24) + d.getLocalTime());`
- D. 35. `DateFormat df= DateFormat.getDateInstance();`
42. `d.setLocalTime((60 * 60 * 24) + d.getLocalTime());`

Answer: B

115. Program

_Given a valid `DateFormat` object named `df`, and

```
16. Date d = new Date(0L);
17. String ds = "December 15, 2004";
18. // insert code here
```

What updates `d`'s value with the date represented by `ds`?

- A. 18. `d = df.parse(ds);`
- B. 18. `d = df.getDate(ds);`
- C. 18. `try {`


```
19. d = df.parse(ds);  
20. } catch(ParseException e) { };  
D. 18. try {  
19. d = df.getDate(ds);  
20. } catch(ParseException e) { };
```

Answer: C

LARA