Oracle_RealExamQuestions.Com_1Z0-851_v2011-11-09_265q_By-Bomas

Exam A

QUESTION 1
A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of `add(0, object)`, but does NOT need to support quick random access. What supports these requirements?

A. `java.util.Queue`
B. `java.util.ArrayList`
C. `java.util.LinearList`
D. `java.util.LinkedList`

Correct Answer: D

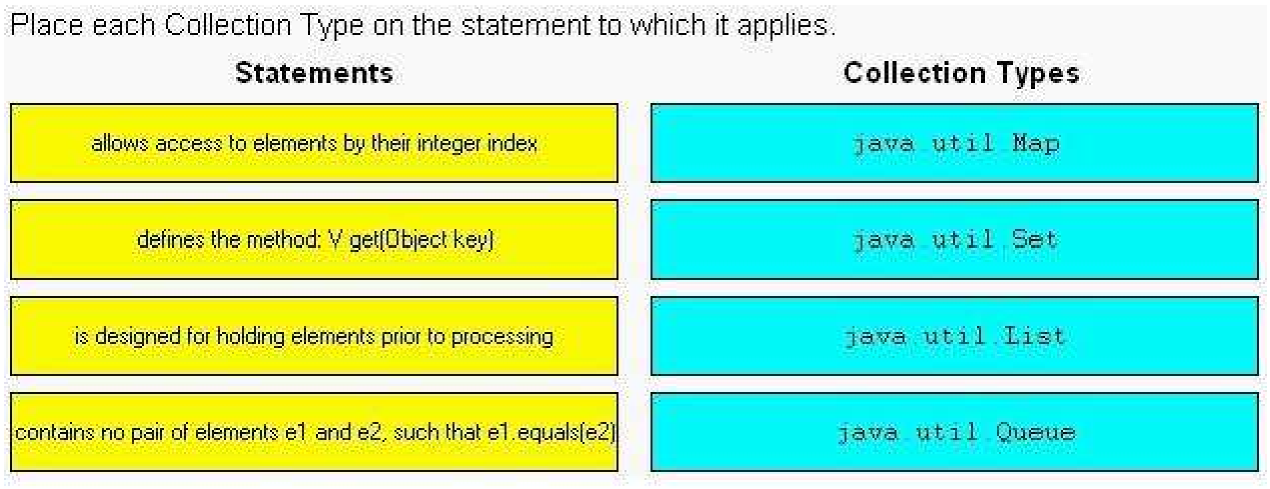
QUESTION 2
Which collection class(es) allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

A. java.util.HashSet
B. java.util.LinkedHashSet
C. java.util.List
D. java.util.ArrayList
E. java.util.Vector
F. java.util.PriorityQueue

Correct Answer: D

QUESTION 3

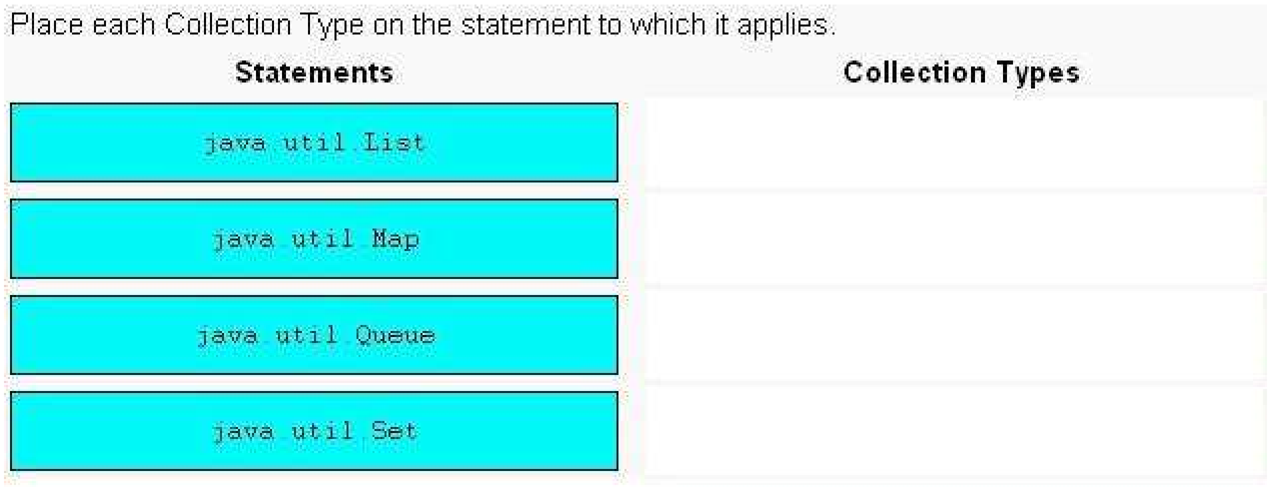
Select and Place:



Place each Collection Type on the statement to which it applies.

Statements	Collection Types
allows access to elements by their integer index	java.util.Map
defines the method: V get(Object key)	java.util.Set
is designed for holding elements prior to processing	java.util.List
contains no pair of elements e1 and e2, such that e1.equals(e2)	java.util.Queue

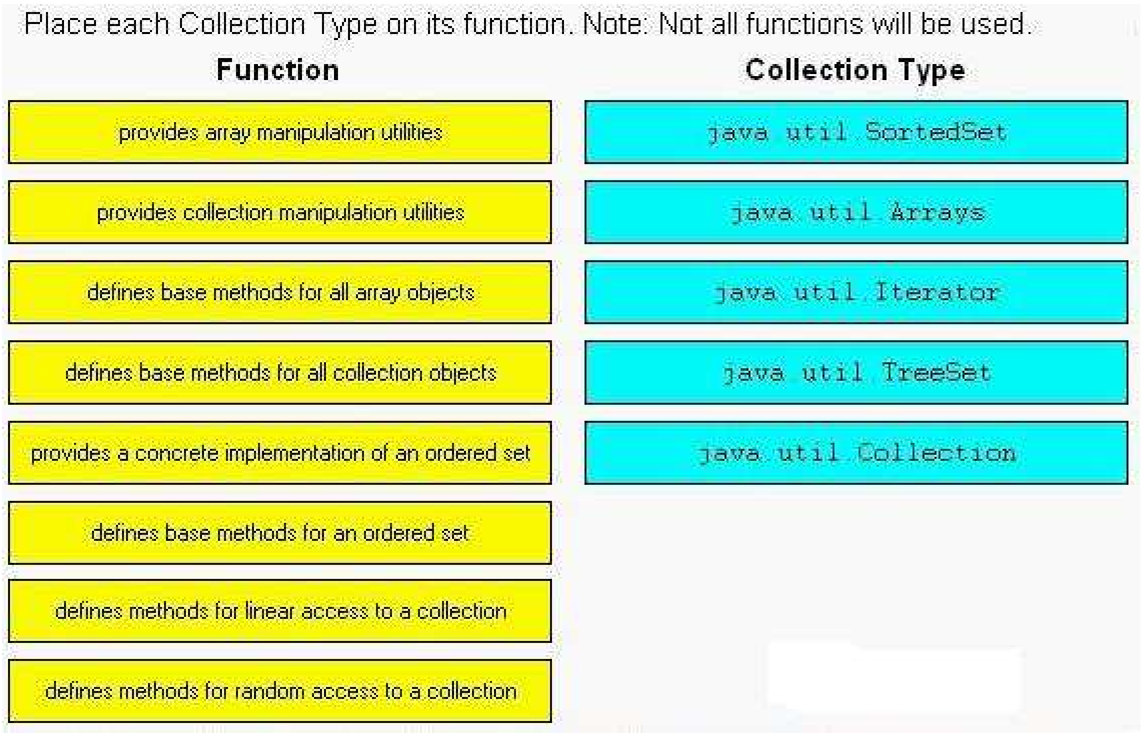
Correct Answer:


Place each Collection Type on the statement to which it applies.

Statements	Collection Types
java.util.List	
java.util.Map	
java.util.Queue	
java.util.Set	

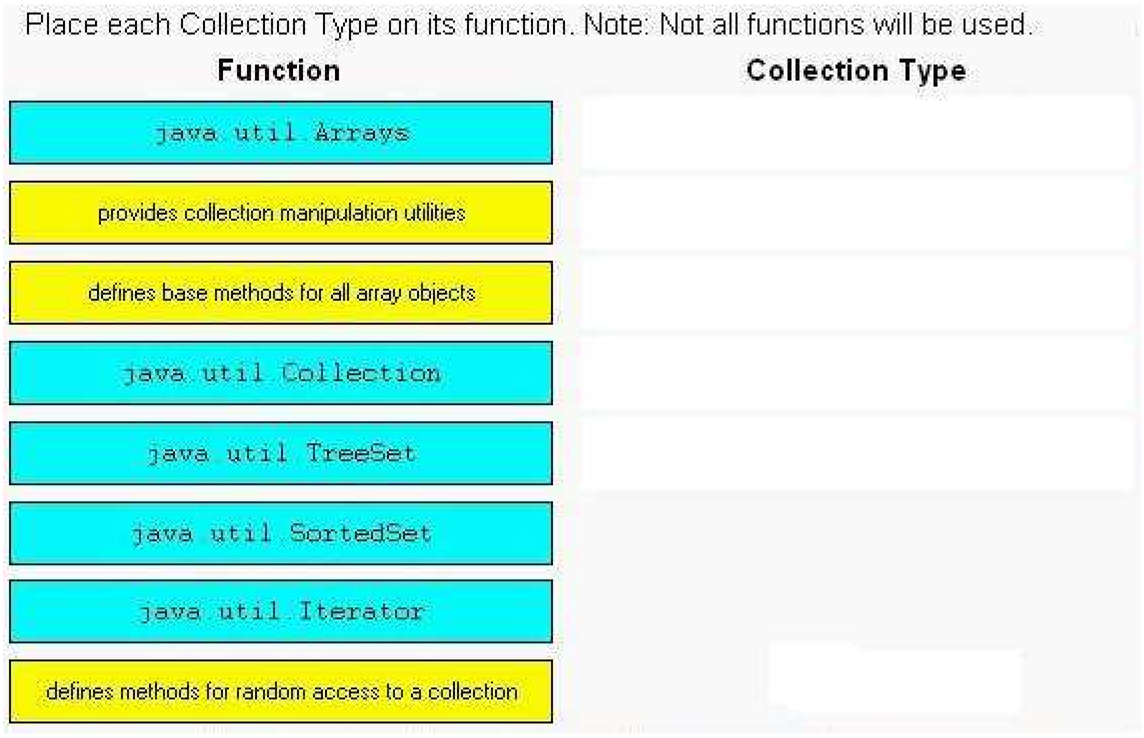
QUESTION 4

Select and Place:


Place each Collection Type on its function. Note: Not all functions will be used.

Function	Collection Type
provides array manipulation utilities	java.util.SortedSet
provides collection manipulation utilities	java.util.Arrays
defines base methods for all array objects	java.util.Iterator
defines base methods for all collection objects	java.util.TreeSet
provides a concrete implementation of an ordered set	java.util.Collection
defines base methods for an ordered set	
defines methods for linear access to a collection	
defines methods for random access to a collection	

Correct Answer:


Place each Collection Type on its function. Note: Not all functions will be used.

Function	Collection Type
java.util.Arrays	
provides collection manipulation utilities	
defines base methods for all array objects	
java.util.Collection	
java.util.TreeSet	
java.util.SortedSet	
java.util.Iterator	
defines methods for random access to a collection	

QUESTION 5
Given a class whose instances, when found in a collection of objects, are sorted by using the compareTo() method, which two statements are true? (Choose two.)

A. The class implements java.lang.Comparable.
B. The class implements java.util.Comparator.
C. The interface used to implement sorting allows this class to define only one sort sequence.
D. The interface used to implement sorting allows this class to define many different sort sequences.

Correct Answer: AC

QUESTION 6
Given:

```
import java.util.*;

public class Quest {
    public static void main(String[] args) {
        String[] colors = {"blue", "red", "green", "yellow", "orange"};
        Arrays.sort(colors);
        int s2 = Arrays.binarySearch(colors, "orange");
        int s3 = Arrays.binarySearch(colors, "violet");
        System.out.println(s2 + " " + s3);
    }
}
```

What is the result?

A. 2 -1
B. 2 -4
C. 2 -5
D. 3 -1
E. 3 -4
F. 3 -5
G. Compilation fails.

H. An exception is thrown at runtime.

Correct Answer: C

QUESTION 7
Given:

```
23. Object [] myObjects = {
24.         new Integer(12),
25.         new String("foo"),
26.         new Integer(5),
27.         new Boolean(true)
28. };
29. Arrays.sort(myObjects);
30. for(int i=0; i<myObjects.length; i++) {
31.    System.out.print(myObjects[i].toString());
32.    System.out.print(" ");
33. }
```

What is the result?

A. Compilation fails due to an error in line 23.
B. Compilation fails due to an error in line 29.
C. A ClassCastException occurs in line 29.
D. A ClassCastException occurs in line 31.
E. The value of all four objects prints in natural order.

Correct Answer: C

QUESTION 8
Given:

```
import java.util.*;

public class LetterASort {
    public static void main(String[] args) {
        String[] strings = new String[]{"aAaA", "AaA", "aAa", "AAaa"};
        Arrays.sort(strings);
        for (String s : strings) {
            System.out.print(s + " ");
        }
    }
}
```

What is the result?

A. Compilation fails.
B. aAaA aAa AAaa AaA
C. AAaa AaA aAa aAaA
D. AaA AAaa aAaA aAa
E. aAa AaA aAaA AAaa
F. An exception is thrown at runtime.

Correct Answer: C

QUESTION 9
Given:

```
import java.util.*;

public class LetterASort {
    public static void main(String[] args) {
        ArrayList<String> strings = new ArrayList<String>();
        strings.add("aAaA");
        strings.add("AaA");
        strings.add("aAa");
        strings.add("AAaa");
        Collections.sort(strings);
        for (String s : strings) {
            System.out.print(s + " ");
        }
    }
}
```

What is the result?

A. Compilation fails.
B. aAaA aAa AAaa AaA
C. AAaa AaA aAa aAaA
D. AaA AAaa aAaA aAa
E. aAa AaA aAaA AAaa
F. An exception is thrown at runtime.

Correct Answer: C

QUESTION 10
Given:

```
11. public void genNumbers() {
12.        ArrayList numbers = new ArrayList();
13.        for (int i = 0; i < 10; i++) {
14.            int value = i * ((int) Math.random());
15.            Integer intObj = new Integer(value);
16.            numbers.add(intObj);
17.        }
18.        System.out.println(numbers);
19. }
```

Which line of code marks the earliest point that an object referenced by intObj becomes a candidate for garbage collection?

A. Line 16
B. Line 17
C. Line 18
D. Line 19
E. The object is NOT a candidate for garbage collection.

Correct Answer: D

QUESTION 11
Given:

```
import java.util.*;
public class Example {
   public static void main(String[] args) {
        // insert code here
        set.add(new Integer(2));
        set.add(new Integer(1));
        System.out.println(set);
   }
}
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

A. `Set set = new TreeSet();`
B. `Set set = new HashSet();`
C. `Set set = new SortedSet();`
D. `List set = new SortedList();`
E. `Set set = new LinkedHashSet();`

Correct Answer: A

QUESTION 12
Given:

```
import java.util.*;

public class Explorer1 {
    public static void main(String[] args) {
        TreeSet<Integer> s = new TreeSet<Integer>();
        TreeSet<Integer> subs = new TreeSet<Integer>();
        for(int i = 606; i < 613; i++)
            if(i%2 == 0) s.add(i);
        subs = (TreeSet)s.subSet(608, true, 611, true);
        s.add(609);
        System.out.println(s + " " + subs);
    }
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. [608, 609, 610, 612] [608, 610]
D. [608, 609, 610, 612] [608, 609, 610]
E. [606, 608, 609, 610, 612] [608, 610]
F. [606, 608, 609, 610, 612] [608, 609, 610]

Correct Answer: F

QUESTION 13
Given:

```
import java.util.TreeSet;

public class Explorer2 {
    public static void main(String[] args) {
        TreeSet<Integer> s = new TreeSet<Integer>();
        TreeSet<Integer> subs = new TreeSet<Integer>();
        for(int i = 606; i < 613; i++)
            if(i%2 == 0) s.add(i);
        subs = (TreeSet)s.subSet(608, true, 611, true);
        s.add(629);
        System.out.println(s + " " + subs);
    }
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. [608, 610, 612, 629] [608, 610]
D. [608, 610, 612, 629] [608, 610, 629]

E. [606, 608, 610, 612, 629] [608, 610]
F. [606, 608, 610, 612, 629] [608, 610, 629]

Correct Answer: E

QUESTION 14
Given:

```
import java.util.*;

public class Explorer3 {
    public static void main(String[] args) {
        TreeSet<Integer> s = new TreeSet<Integer>();
        TreeSet<Integer> subs = new TreeSet<Integer>();
        for (int i = 606; i < 613; i++)
            if (i % 2 == 0)
                s.add(i);
        subs = (TreeSet) s.subSet(608, true, 611, true);
        subs.add(629);
        System.out.println(s + " " + subs);
    }
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. [608, 610, 612, 629] [608, 610]
D. [608, 610, 612, 629] [608, 610, 629]
E. [606, 608, 610, 612, 629] [608, 610]
F. [606, 608, 610, 612, 629] [608, 610, 629]

Correct Answer: B

QUESTION 15
Given that the elements of a PriorityQueue are ordered according to natural ordering, and:

```
import java.util.*;

public class GetInLine {
    public static void main(String[] args) {
        PriorityQueue<String> pq = new PriorityQueue<String>();
        pq.add("banana");
        pq.add("pear");
        pq.add("apple");
        System.out.println(pq.poll() + " " + pq.peek());
    }
}
```

What is the result?

A. apple pear
B. banana pear
C. apple apple
D. apple banana
E. banana banana

Correct Answer: D

QUESTION 16

Given:

```
import java.util.*;

public class Mapit {
    public static void main(String[] args) {
        Set<Integer> set = new HashSet<Integer>();
        Integer i1 = 45;
        Integer i2 = 46;
        set.add(i1);
        set.add(i1);
        set.add(i2); System.out.print(set.size() + " ");
        set.remove(i1); System.out.print(set.size() + " ");
        i2 = 47;
        set.remove(i2); System.out.print(set.size() + " ");
    }
}
```

What is the result?

A. 2 1 0
B. 2 1 1
C. 3 2 1
D. 3 2 2
E. Compilation fails.
F. An exception is thrown at runtime.

Correct Answer: B

QUESTION 17
Given:

```
import java.util.*;

public class WrappedString {
    private String s;

    public WrappedString(String s) { this.s = s; }

    public static void main(String[] args) {
        HashSet<Object> hs = new HashSet<Object>();
        WrappedString ws1 = new WrappedString("aardvark");
        WrappedString ws2 = new WrappedString("aardvark");
        String s1 = new String("aardvark");
        String s2 = new String("aardvark");
        hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);
        System.out.println(hs.size()); }
}
```

What is the result?

A. 0
B. 1
C. 2
D. 3
E. 4
F. Compilation fails.
G. An exception is thrown at runtime.

Correct Answer: D

QUESTION 18

Given:

```
import java.util.*;

public class SortOf {
    public static void main(String[] args) {
        ArrayList<Integer> a = new ArrayList<Integer>();
        a.add(1); a.add(5); a.add(3);
        Collections.sort(a);
        a.add(2);
        Collections.reverse(a);
        System.out.println(a);
    }
}
```

What is the result?

A. [1, 2, 3, 5]
B. [2, 1, 3, 5]
C. [2, 5, 3, 1]
D. [5, 3, 2, 1]
E. [1, 3, 5, 2]
F. Compilation fails.
G. An exception is thrown at runtime.

Correct Answer: C

QUESTION 19
Given:

```
public static Collection get() {
    Collection sorted = new LinkedList();
    sorted.add("B"); sorted.add("C"); sorted.add("A");
    return sorted;
}

public static void main(String[] args) {
    for (Object obj: get()) {
        System.out.print(obj + ", ");
    }
}
```

What is the result?

A. A, B, C,
B. B, C, A,
C. Compilation fails.
D. The code runs with no output.
E. An exception is thrown at runtime.

Correct Answer: B

QUESTION 20
Given:

```
34. HashMap props = new HashMap();
35. props.put("key45", "some value");
36. props.put("key12", "some other value");
37. props.put("key39", "yet another value");
38. Set s = props.keySet();
39. //insert code here
```

What, inserted at line 39, will sort the keys in the props HashMap?

A. `Arrays.sort(s);`
B. `s = new TreeSet(s);`
C. `Collections.sort(s);`
D. `s = new SortedSet(s);`

Correct Answer: B

QUESTION 21
Given:

```
public static Iterator reverse(List list) {
    Collections.reverse(list);
    return list.iterator();
}

public static void main(String[] args) {
    List list = new ArrayList();
    list.add("1"); list.add("2"); list.add("3");
    for (Object obj: reverse(list))
        System.out.print(obj + ", ");
}
```

What is the result?

A. 3, 2, 1,
B. 1, 2, 3,
C. Compilation fails.
D. The code runs with no output.
E. An exception is thrown at runtime.

Correct Answer: C

QUESTION 22
Given:

```
class Pizza {
    java.util.ArrayList toppings;

    public final void addTopping(String topping) {
        toppings.add(topping);
    }
    public void removeTopping(String topping) {
        toppings.remove(topping);
    }
}

public class PepperoniPizza extends Pizza {
    public void addTopping(String topping) {
        System.out.println("Cannot add Toppings");
    }

    public static void main(String[] args) {
        Pizza pizza = new PepperoniPizza();
        pizza.addTopping("Mushrooms");
        pizza.removeTopping("Peperoni");
    }
}
```

What is the result?

A. Compilation fails.

B. Cannot add Toppings
C. The code runs with no output.
D. A NullPointerException is thrown in Line 4.

Correct Answer: A

QUESTION 23
Given:

```
interface A { void x(); }
class B implements A { public void x() {} public void y() {} }
class C extends B { public void x() {} }
```

and:

```
20. java.util.List<A> list = new java.util.ArrayList<A>();
21. list.add(new B());
22. list.add(new C());
23. for (A a : list) {
24.    a.x();
25.    a.y();
26. }
```

What is the result?

A. The code runs with no output.
B. An exception is thrown at runtime.
C. Compilation fails because of an error in line 20.
D. Compilation fails because of an error in line 21.
E. Compilation fails because of an error in line 23.
F. Compilation fails because of an error in line 25.

Correct Answer: F

QUESTION 24
Given a pre-generics implementation of a method:

```
11. public static int sum(List list) {
12.    int sum = 0;
13.    for ( Iterator iter = list.iterator(); iter.hasNext(); ) {
14.        int i = ((Integer)iter.next()).intValue();
15.        sum += i;
16.    }
17.    return sum;
18. }
```

What three changes allow the class to be used with generics and avoid an unchecked warning? (Choose three.)

A. Remove line 14.
B. Replace line 14 with `int i = iter.next();`
C. Replace line 13 with `for (int i : intList) {`
D. Replace line 13 with `for (Iterator iter : intList) {`
E. Replace the method declaration `with sum(List<int> intList)`
F. Replace the method declaration `with sum(List<Integer> intList)`

Correct Answer: ACF

QUESTION 25
Given:

```
11. //insert code here
12.   private N min, max;
13.   public N getMin() { return min; }
14.   public N getMax() { return max; }
15.   public void add(N added) {
16.      if (min == null || added.doubleValue() < min.doubleValue())
17.         min = added;
18.      if (max == null || added.doubleValue() > max.doubleValue())
19         max = added;
20.   }
21. }
```

Which two, inserted at line 11, will allow the code to compile? (Choose two.)

A. `public class MinMax<?> {`
B. `public class MinMax<? extends Number> {`
C. `public class MinMax<N extends Object> {`
D. `public class MinMax<N extends Number> {`
E. `public class MinMax<? extends Object> {`
F. `public class MinMax<N extends Integer> {`

Correct Answer: DF

QUESTION 26
Given:

```
3. import java.util.*;
4. public class G1 {
5.    public void takeList(List<? extends String> list) {
6.       // insert code here
7.    }
8. }
```

Which three code fragments, inserted independently at line 6, will compile? (Choose three.)

A. `list.add("foo");`
B. `Object o = list;`
C. `String s = list.get(0);`
D. `list = new ArrayList<String>();`
E. `list = new ArrayList<Object>();`

Correct Answer: BCD

QUESTION 27
A programmer must create a generic class MinMax and the type parameter of MinMax must implement Comparable. Which implementation of MinMax will compile?

A.
```
class MinMax<E extends Comparable<E>> {
      E min = null;
      E max = null;
      public MinMax() {}
      public void put(E value) { /* store min or max */ }
   }
```
B.
```
class MinMax<E implements Comparable<E>> {
      E min = null;
      E max = null;
      public MinMax() {}
      public void put(E value) { /* store min or max */ }
   }
```
C.
```
class MinMax<E extends Comparable<E>> {
      <E> E min = null;
```

```
       <E> E max = null;
       public MinMax() {}
       public <E> void put(E value) { /* store min or max */ }
   }
```

D.
```
class MinMax<E implements Comparable<E>> {
       <E> E min = null;
       <E> E max = null;
       public MinMax() {}
       public <E> void put(E value) { /* store min or max */ }
   }
```

Correct Answer: A

QUESTION 28
Given:

```
03. import java.util.*;
04. public class Hancock {
05.        // insert code here
06.        list.add("foo");
07.    }
08. }
```

Which two code fragments, inserted independently at line 5, will compile without warnings? (Choose two.)

A. `public void addStrings(List list) {`
B. `public void addStrings(List<String> list) {`
C. `public void addStrings(List<? super String> list) {`
D. `public void addStrings(List<? extends String> list) {`

Correct Answer: BC

QUESTION 29
Given:

```
public class Drink implements Comparable {
    public String name;
    public int compareTo(Object o) {
        return 0;
    }
}
```

and:

```
Drink one = new Drink();
Drink two = new Drink();
one.name= "Coffee";
two.name= "Tea";
TreeSet set = new TreeSet();
set.add(one);
set.add(two);
```

A programmer iterates over the TreeSet and prints the name of each Drink object. What is the result?

A. Tea
B. Coffee
C. Coffee
   Tea
D. Compilation fails.
E. The code runs with no output.
F. An exception is thrown at runtime.

Correct Answer: B

QUESTION 30
Given:

```
public class Person {
    private name;
    public Person(String name) {
        this.name = name;
    }
    public int hashCode() {
        return 420;
    }
}
```

Which statement is true?

A. The time to find the value from HashMap with a Person key depends on the size of the map.
B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.
C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.
D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

Correct Answer: A

QUESTION 31
Given:

```
public class Key {
    private long id1;
    private long id2;

    // class Key methods
}
```

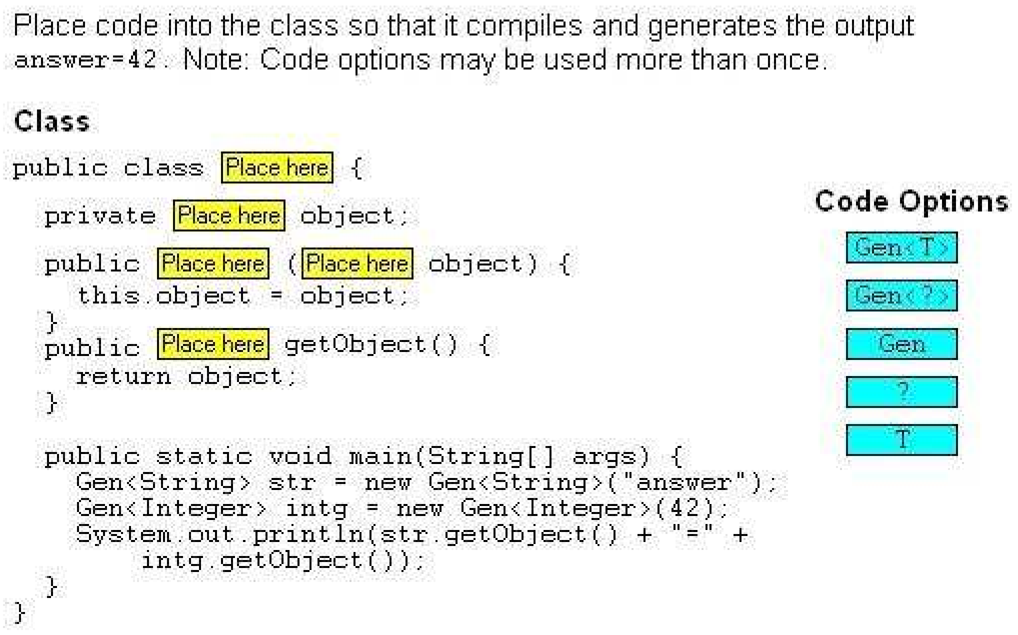
A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap. Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

A. `public int hashCode()`
B. `public boolean equals(Key k)`
C. `public int compareTo(Object o)`
D. `public boolean equals(Object o)`
E. `public boolean compareTo(Key k)`

Correct Answer: AD

QUESTION 32

Select and Place:



Place code into the class so that it compiles and generates the output `answer=42`. Note: Code options may be used more than once.

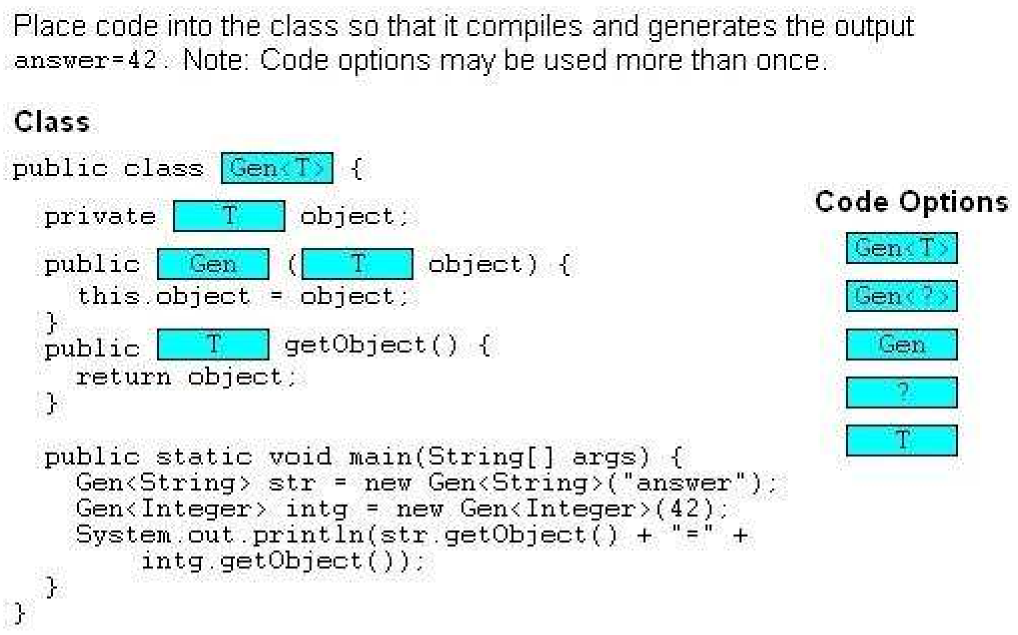
Class

```
public class [Place here] {
  private [Place here] object;
  public [Place here] ([Place here] object) {
    this.object = object;
  }
  public [Place here] getObject() {
    return object;
  }

  public static void main(String[] args) {
    Gen<String> str = new Gen<String>("answer");
    Gen<Integer> intg = new Gen<Integer>(42);
    System.out.println(str.getObject() + "=" +
        intg.getObject());
  }
}
```

Code Options

- Gen<T>
- Gen<?>
- Gen
- ?
- T

Correct Answer:



Place code into the class so that it compiles and generates the output `answer=42`. Note: Code options may be used more than once.

Class

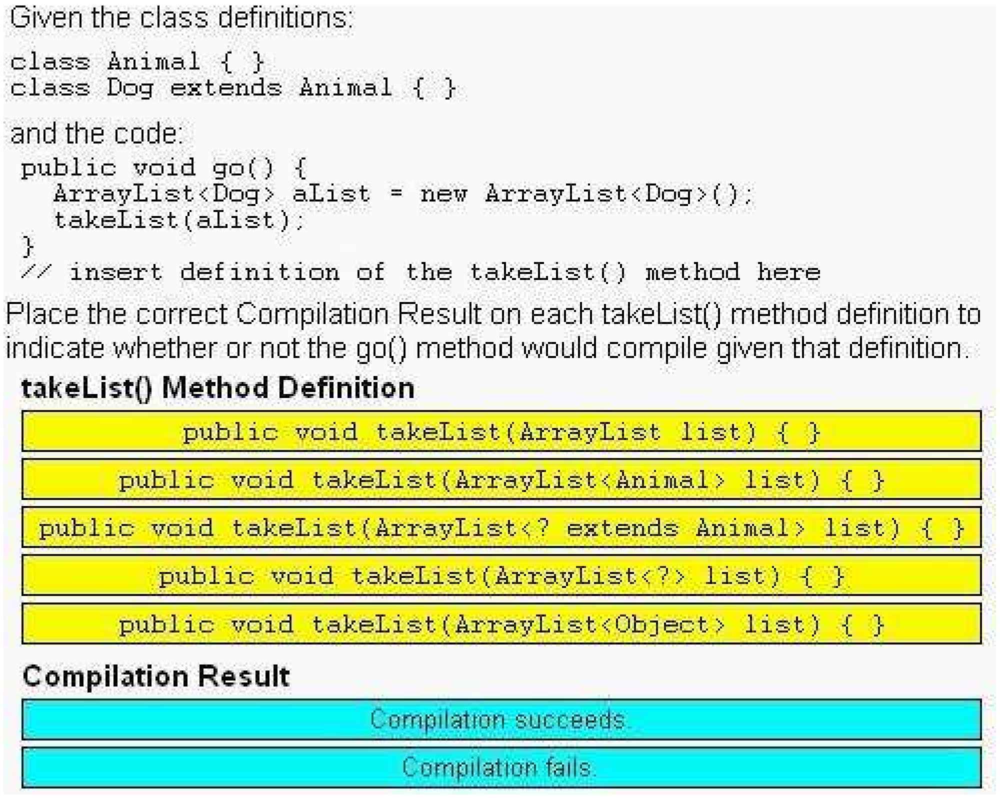
```
public class Gen<T> {
  private T object;
  public Gen (T object) {
    this.object = object;
  }
  public T getObject() {
    return object;
  }

  public static void main(String[] args) {
    Gen<String> str = new Gen<String>("answer");
    Gen<Integer> intg = new Gen<Integer>(42);
    System.out.println(str.getObject() + "=" +
        intg.getObject());
  }
}
```

Code Options

- Gen<T>
- Gen<?>
- Gen
- ?
- T

QUESTION 33

Select and Place:



Given the class definitions:

```
class Animal { }
class Dog extends Animal { }
```

and the code:

```
public void go() {
  ArrayList<Dog> aList = new ArrayList<Dog>();
  takeList(aList);
}
// insert definition of the takeList() method here
```

Place the correct Compilation Result on each takeList() method definition to indicate whether or not the go() method would compile given that definition.

takeList() Method Definition

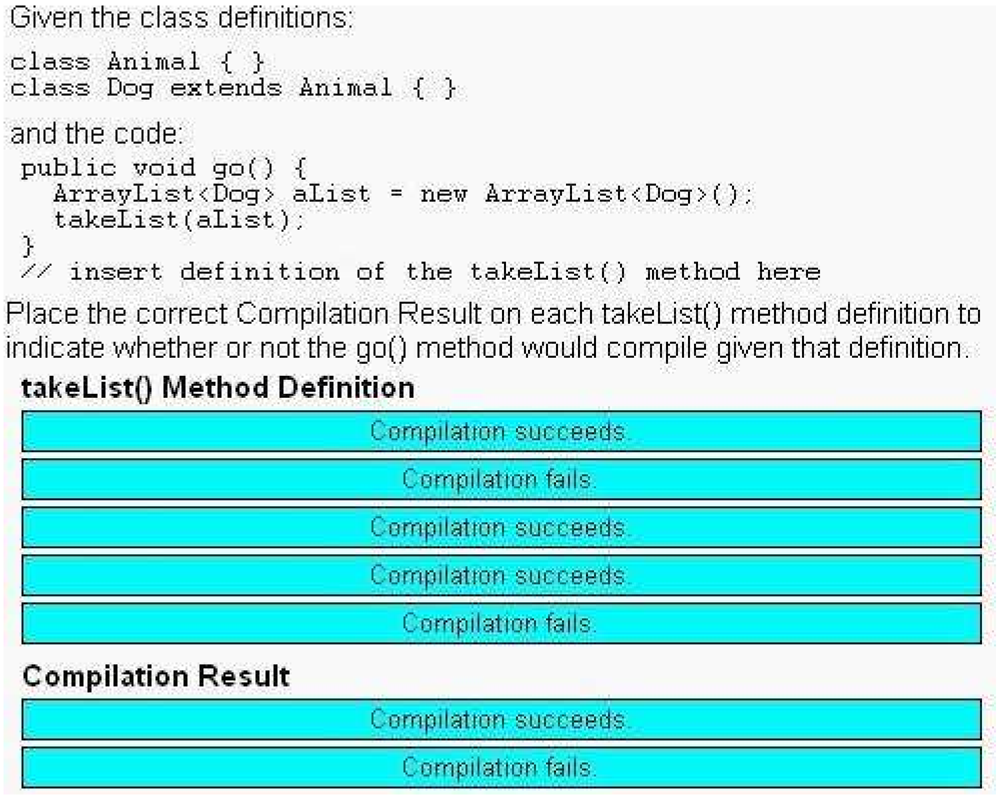
```
public void takeList(ArrayList list) { }
public void takeList(ArrayList<Animal> list) { }
public void takeList(ArrayList<? extends Animal> list) { }
public void takeList(ArrayList<?> list) { }
public void takeList(ArrayList<Object> list) { }
```

Compilation Result

Compilation succeeds.

Compilation fails.

Correct Answer:



Given the class definitions:

```
class Animal { }
class Dog extends Animal { }
```

and the code:

```
public void go() {
  ArrayList<Dog> aList = new ArrayList<Dog>();
  takeList(aList);
}
// insert definition of the takeList() method here
```

Place the correct Compilation Result on each takeList() method definition to indicate whether or not the go() method would compile given that definition.

takeList() Method Definition

Compilation succeeds.

Compilation fails.

Compilation succeeds.

Compilation succeeds.

Compilation fails.

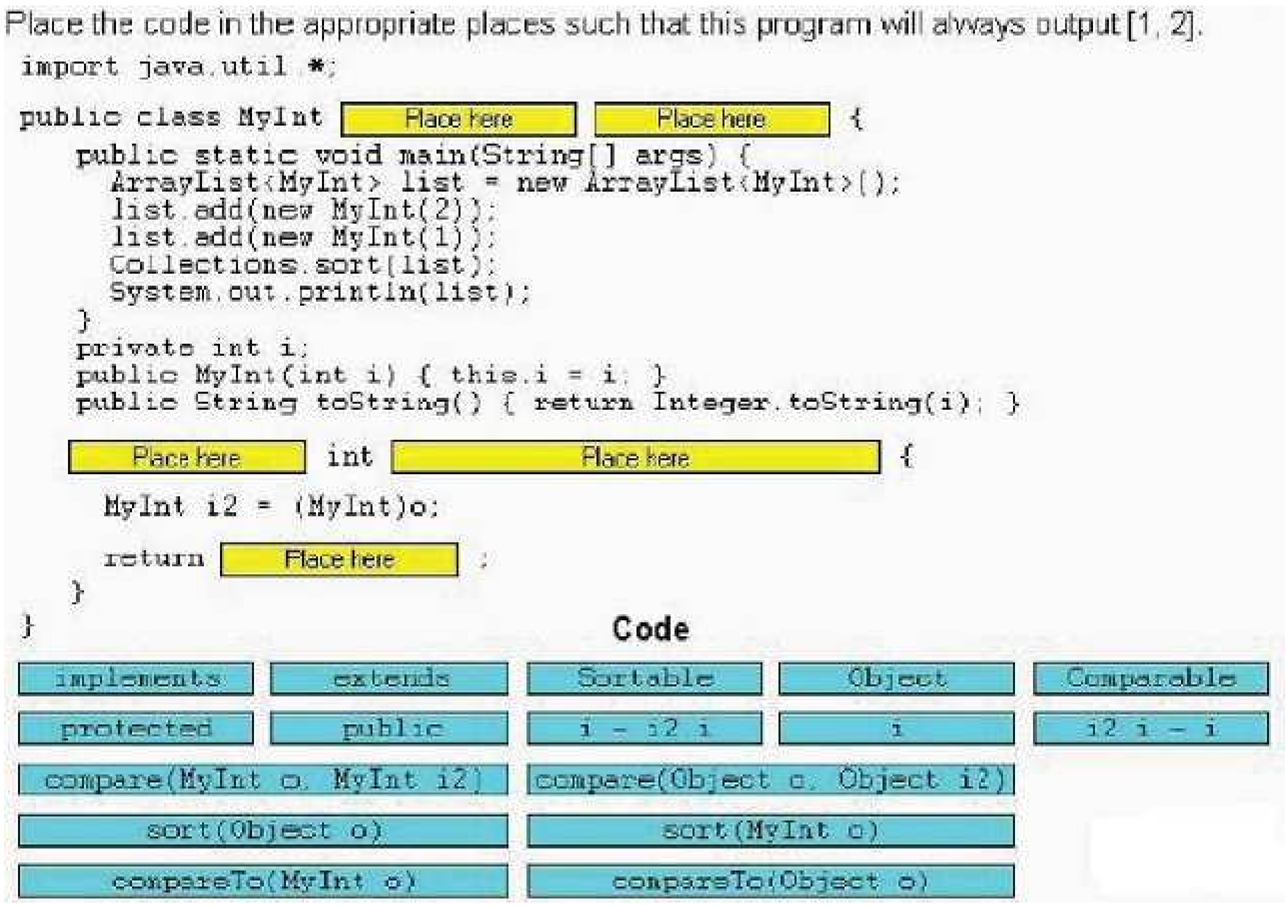
Compilation Result

Compilation succeeds.

Compilation fails.

QUESTION 34

Select and Place:



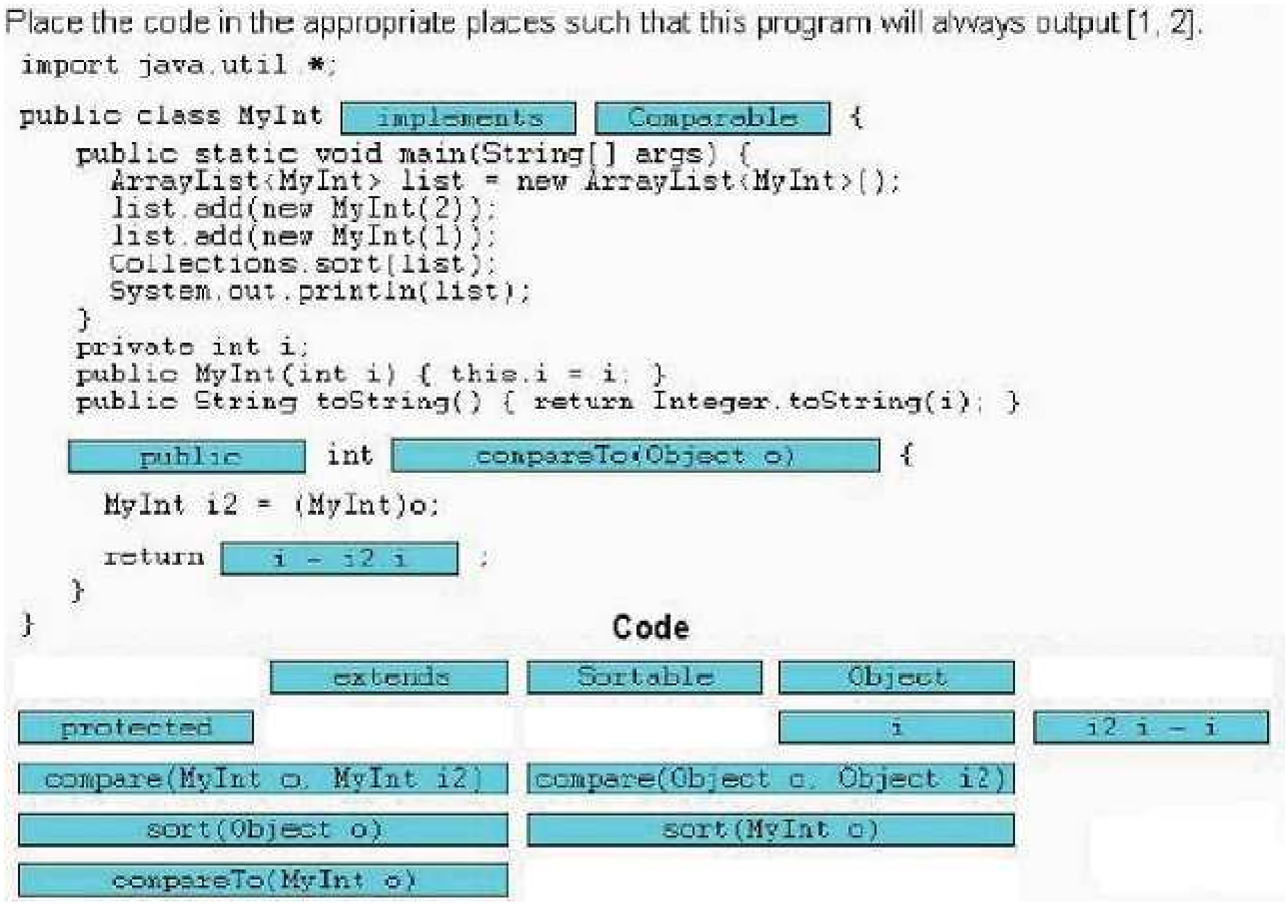
Place the code in the appropriate places such that this program will always output [1, 2].

```
import java.util.*;

public class MyInt  Place here   Place here  {
    public static void main(String[] args) {
      ArrayList<MyInt> list = new ArrayList<MyInt>();
      list.add(new MyInt(2));
      list.add(new MyInt(1));
      Collections.sort(list);
      System.out.println(list);
    }
    private int i;
    public MyInt(int i) { this.i = i; }
    public String toString() { return Integer.toString(i); }

   Place here   int   Place here   {
     MyInt i2 = (MyInt)o;

     return   Place here   ;
   }
}
```

Code

implements	extends	Sortable	Object	Comparable
protected	public	i - i2.i	i	i2.i - i
compare(MyInt o, MyInt i2)		compare(Object o, Object i2)		
sort(Object o)		sort(MyInt o)		
compareTo(MyInt o)		compareTo(Object o)		

Correct Answer:



Place the code in the appropriate places such that this program will always output [1, 2].

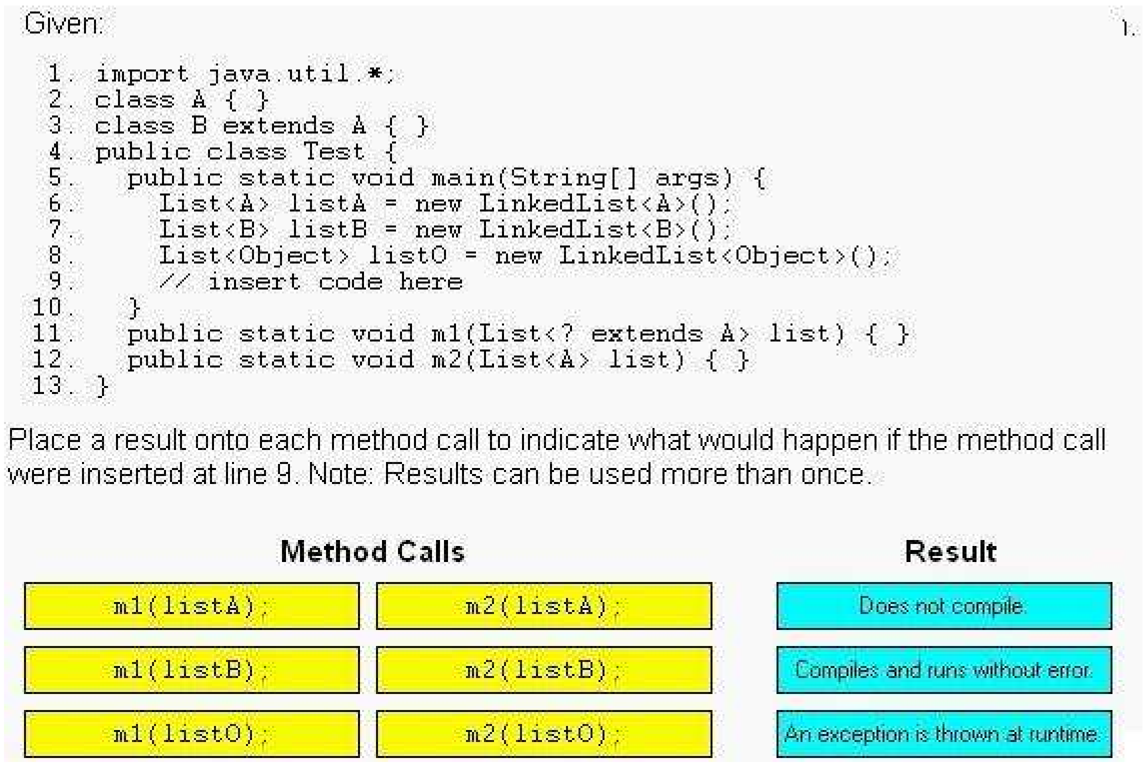
```
import java.util.*;

public class MyInt  implements   Comparable  {
    public static void main(String[] args) {
      ArrayList<MyInt> list = new ArrayList<MyInt>();
      list.add(new MyInt(2));
      list.add(new MyInt(1));
      Collections.sort(list);
      System.out.println(list);
    }
    private int i;
    public MyInt(int i) { this.i = i; }
    public String toString() { return Integer.toString(i); }

   public   int   compareTo(Object o)   {
     MyInt i2 = (MyInt)o;

     return   i - i2.i   ;
   }
}
```

Code

	extends	Sortable	Object	
protected			i	i2.i - i
compare(MyInt o, MyInt i2)		compare(Object o, Object i2)		
sort(Object o)		sort(MyInt o)		
compareTo(MyInt o)				

QUESTION 35

Select and Place:



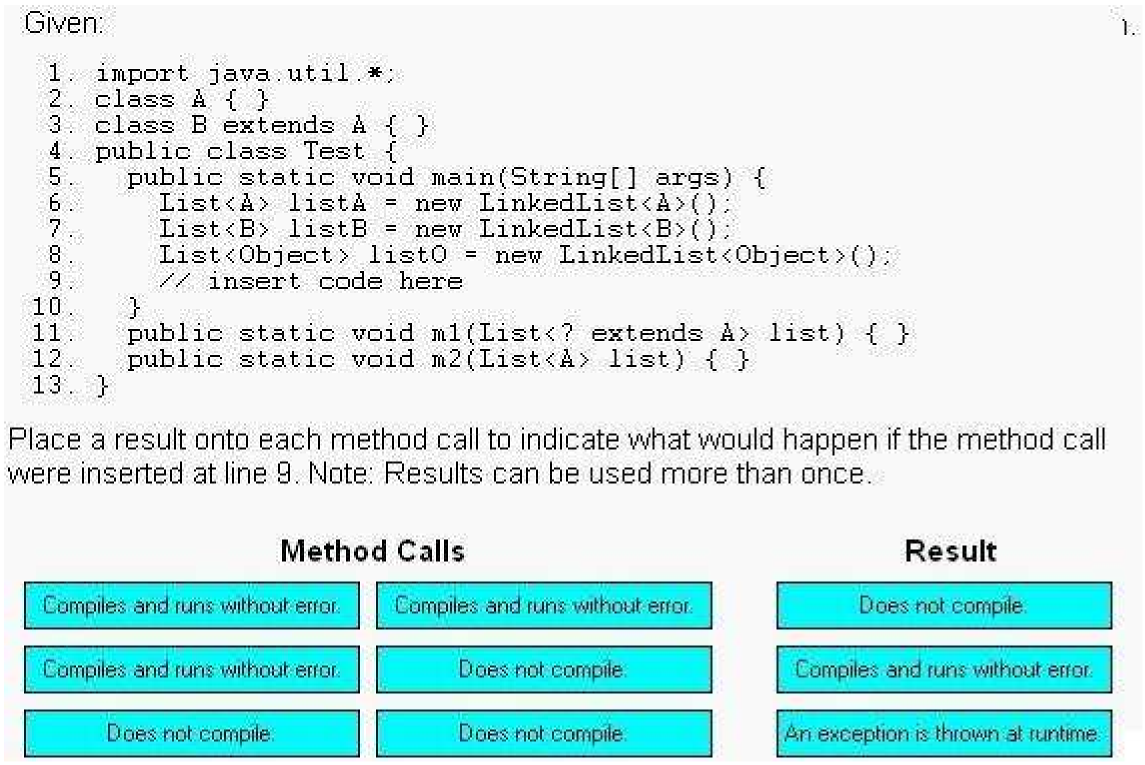
Given:

```
 1. import java.util.*;
 2. class A { }
 3. class B extends A { }
 4. public class Test {
 5.   public static void main(String[] args) {
 6.     List<A> listA = new LinkedList<A>();
 7.     List<B> listB = new LinkedList<B>();
 8.     List<Object> listO = new LinkedList<Object>();
 9.     // insert code here
10.   }
11.   public static void m1(List<? extends A> list) { }
12.   public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
m1(listA);	m2(listA);	Does not compile.
m1(listB);	m2(listB);	Compiles and runs without error.
m1(listO);	m2(listO);	An exception is thrown at runtime.

Correct Answer:



Given:

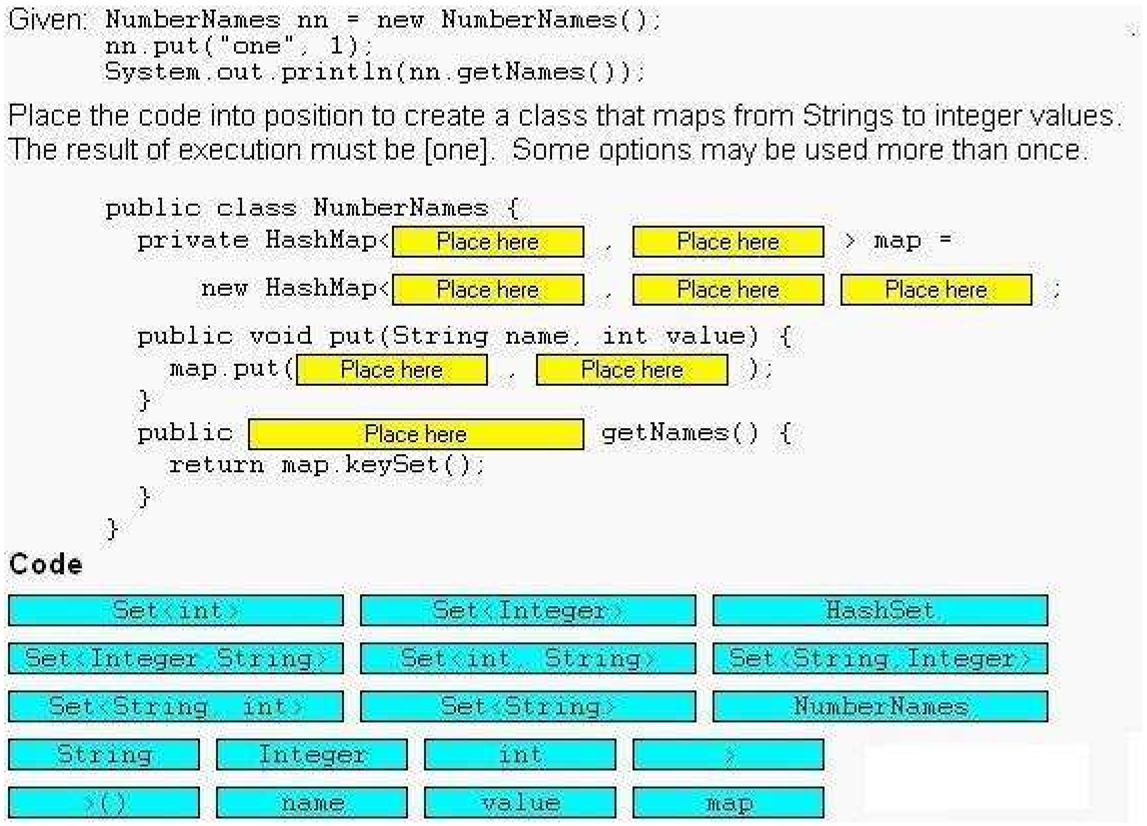
```
 1. import java.util.*;
 2. class A { }
 3. class B extends A { }
 4. public class Test {
 5.   public static void main(String[] args) {
 6.     List<A> listA = new LinkedList<A>();
 7.     List<B> listB = new LinkedList<B>();
 8.     List<Object> listO = new LinkedList<Object>();
 9.     // insert code here
10.   }
11.   public static void m1(List<? extends A> list) { }
12.   public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
Compiles and runs without error.	Compiles and runs without error.	Does not compile.
Compiles and runs without error.	Does not compile.	Compiles and runs without error.
Does not compile.	Does not compile.	An exception is thrown at runtime.

QUESTION 36

Select and Place:



Given:
```
NumberNames nn = new NumberNames();
nn.put("one", 1);
System.out.println(nn.getNames());
```

Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

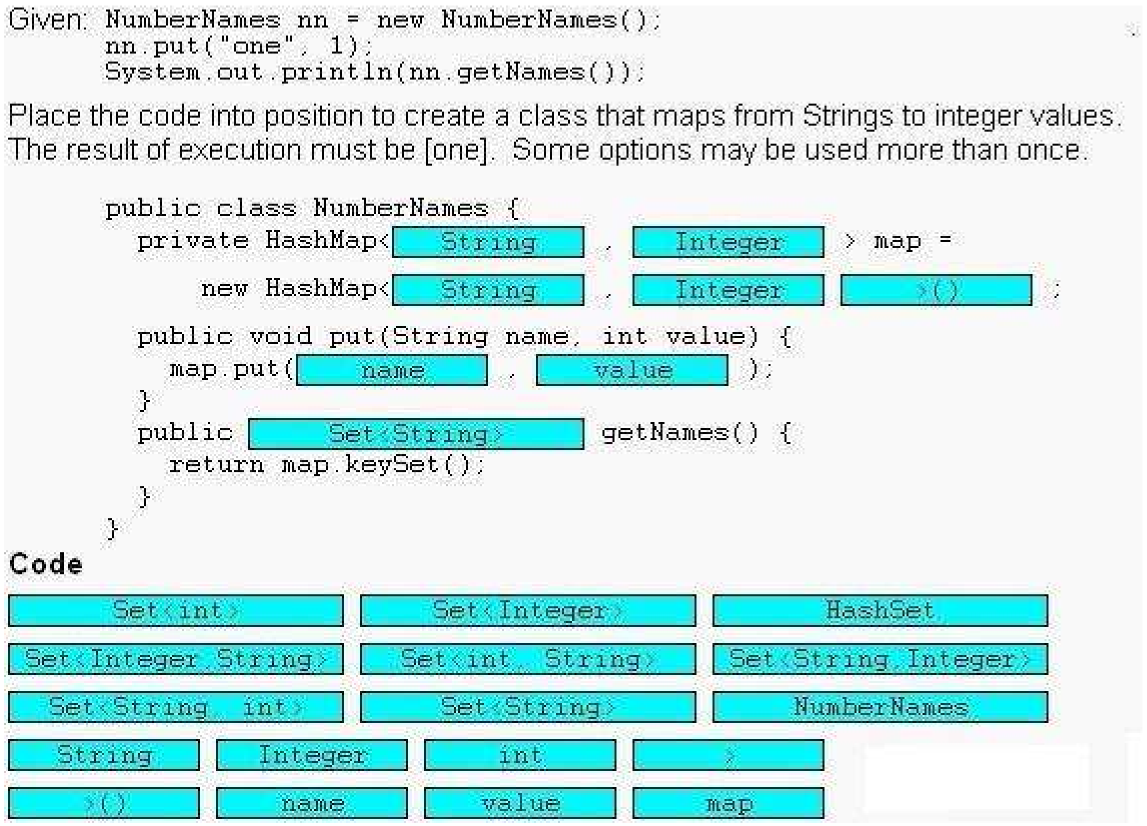
```
public class NumberNames {
  private HashMap<[Place here] , [Place here] > map =
      new HashMap<[Place here] , [Place here] [Place here] ;
  public void put(String name, int value) {
    map.put([Place here] , [Place here] );
  }
  public [Place here] getNames() {
    return map.keySet();
  }
}
```

Code

Set<int>	Set<Integer>	HashSet
Set<Integer,String>	Set<int, String>	Set<String,Integer>
Set<String, int>	Set<String>	NumberNames

String	Integer	int	>
>()	name	value	map

Correct Answer:



Given:
```
NumberNames nn = new NumberNames();
nn.put("one", 1);
System.out.println(nn.getNames());
```

Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

```
public class NumberNames {
  private HashMap<String , Integer > map =
      new HashMap<String , Integer >() ;
  public void put(String name, int value) {
    map.put(name , value );
  }
  public Set<String> getNames() {
    return map.keySet();
  }
}
```

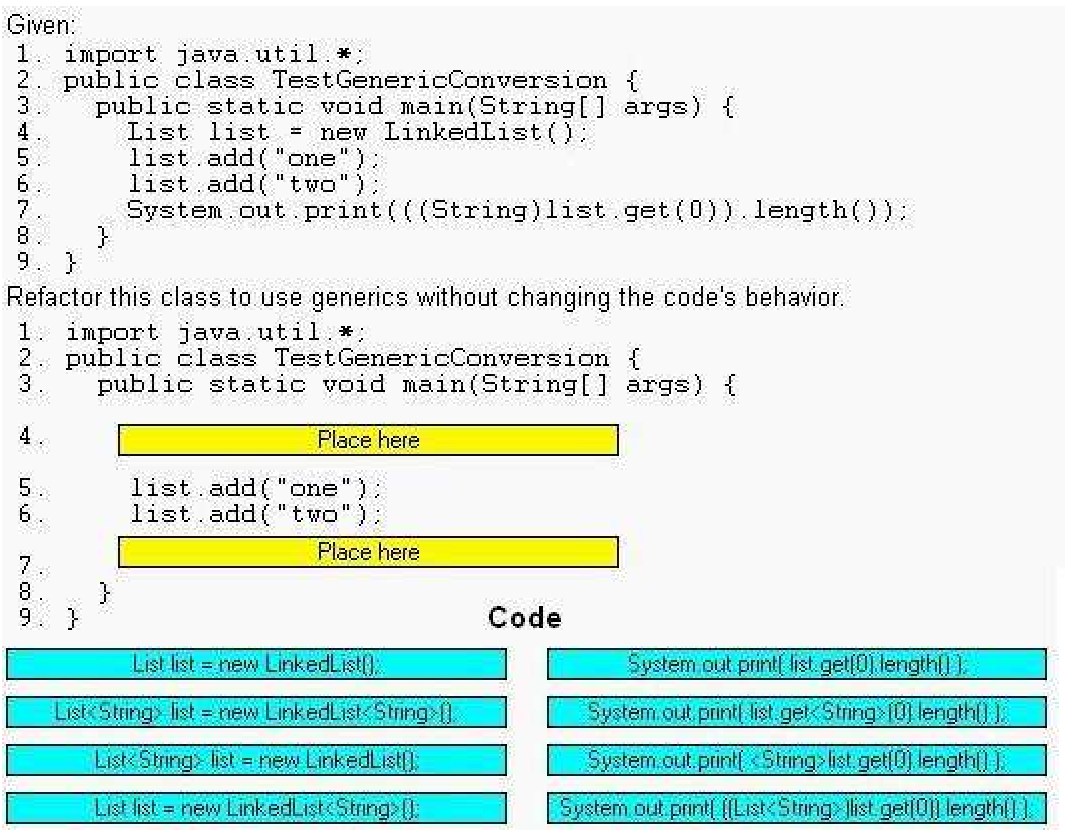
Code

Set<int>	Set<Integer>	HashSet
Set<Integer,String>	Set<int, String>	Set<String,Integer>
Set<String, int>	Set<String>	NumberNames

String	Integer	int	>
>()	name	value	map

QUESTION 37

Select and Place:



Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3.   public static void main(String[] args) {
4.     List list = new LinkedList();
5.     list.add("one");
6.     list.add("two");
7.     System.out.print(((String)list.get(0)).length());
8.   }
9. }
```

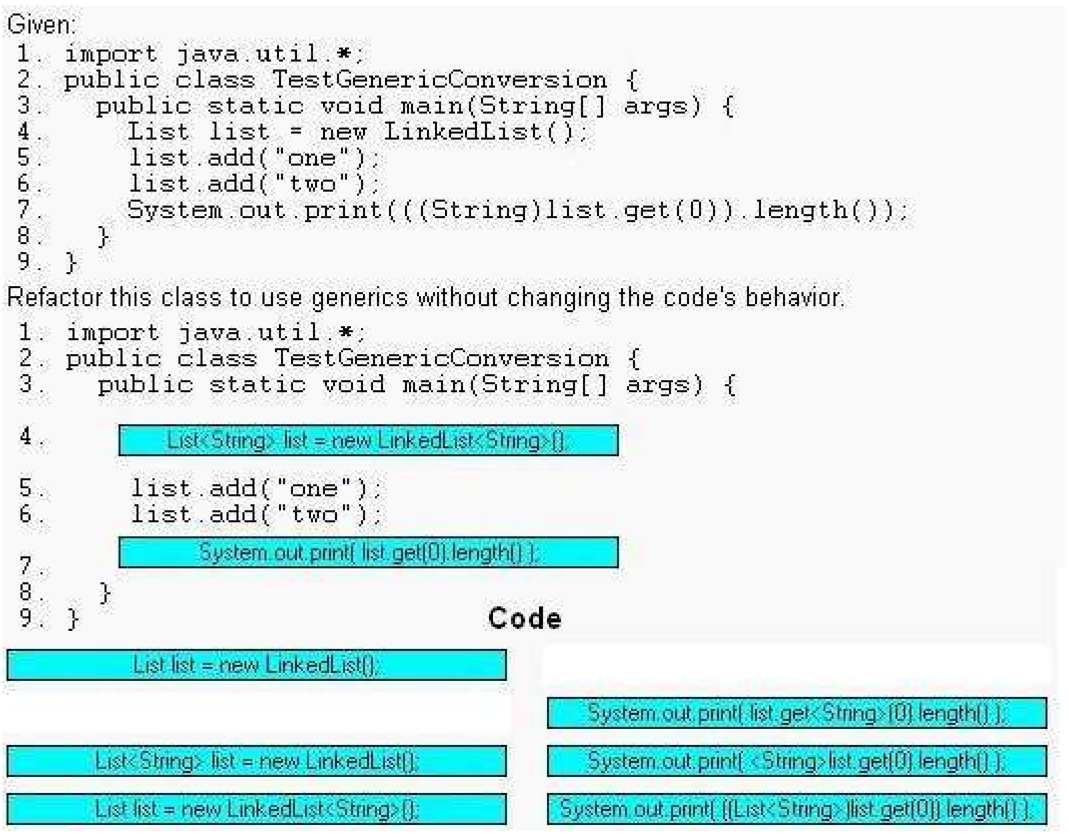
Refactor this class to use generics without changing the code's behavior.

```
1. import java.util.*;
2. public class TestGenericConversion {
3.   public static void main(String[] args) {
4.     Place here
5.     list.add("one");
6.     list.add("two");
7.     Place here
8.   }
9. }
```

Code

List list = new LinkedList();	System.out.print(list.get(0).length());
List<String> list = new LinkedList<String>();	System.out.print(list.get<String>(0).length());
List<String> list = new LinkedList();	System.out.print(<String>list.get(0).length());
List list = new LinkedList<String>();	System.out.print(((List<String>)list.get(0)).length());

Correct Answer:



Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3.   public static void main(String[] args) {
4.     List list = new LinkedList();
5.     list.add("one");
6.     list.add("two");
7.     System.out.print(((String)list.get(0)).length());
8.   }
9. }
```

Refactor this class to use generics without changing the code's behavior.

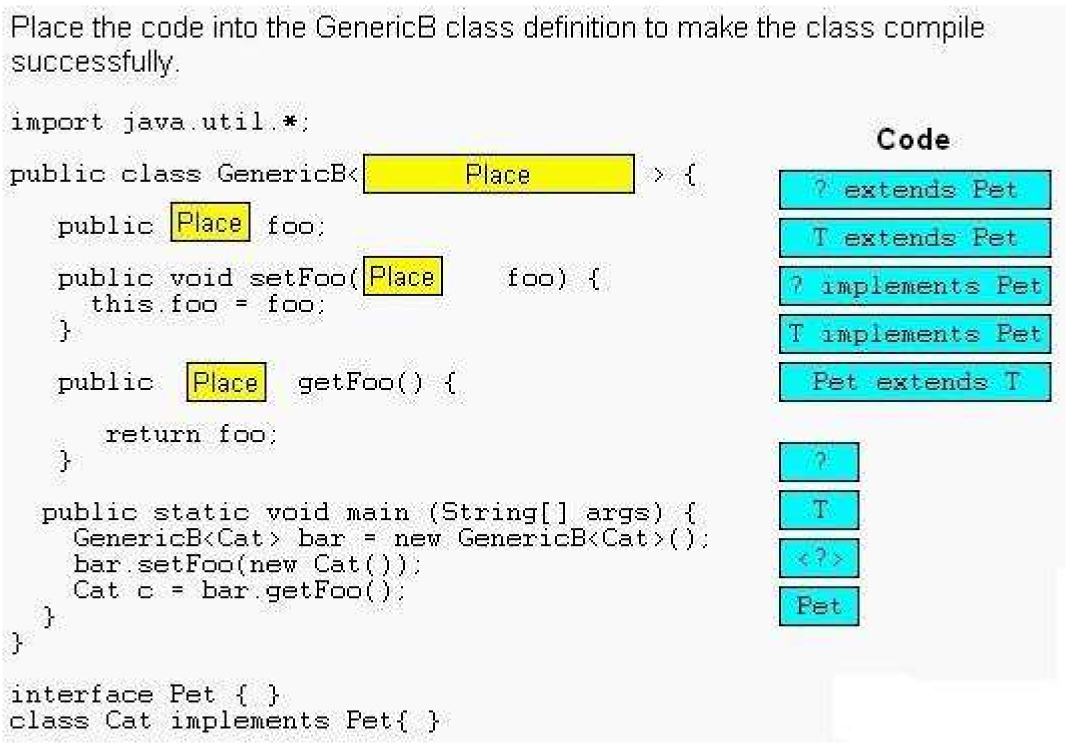
```
1. import java.util.*;
2. public class TestGenericConversion {
3.   public static void main(String[] args) {
4.     List<String> list = new LinkedList<String>();
5.     list.add("one");
6.     list.add("two");
7.     System.out.print( list.get(0).length() );
8.   }
9. }
```

Code

List list = new LinkedList();	
	System.out.print(list.get<String>(0).length());
List<String> list = new LinkedList();	System.out.print(<String>list.get(0).length());
List list = new LinkedList<String>();	System.out.print(((List<String>)list.get(0)).length());

QUESTION 38

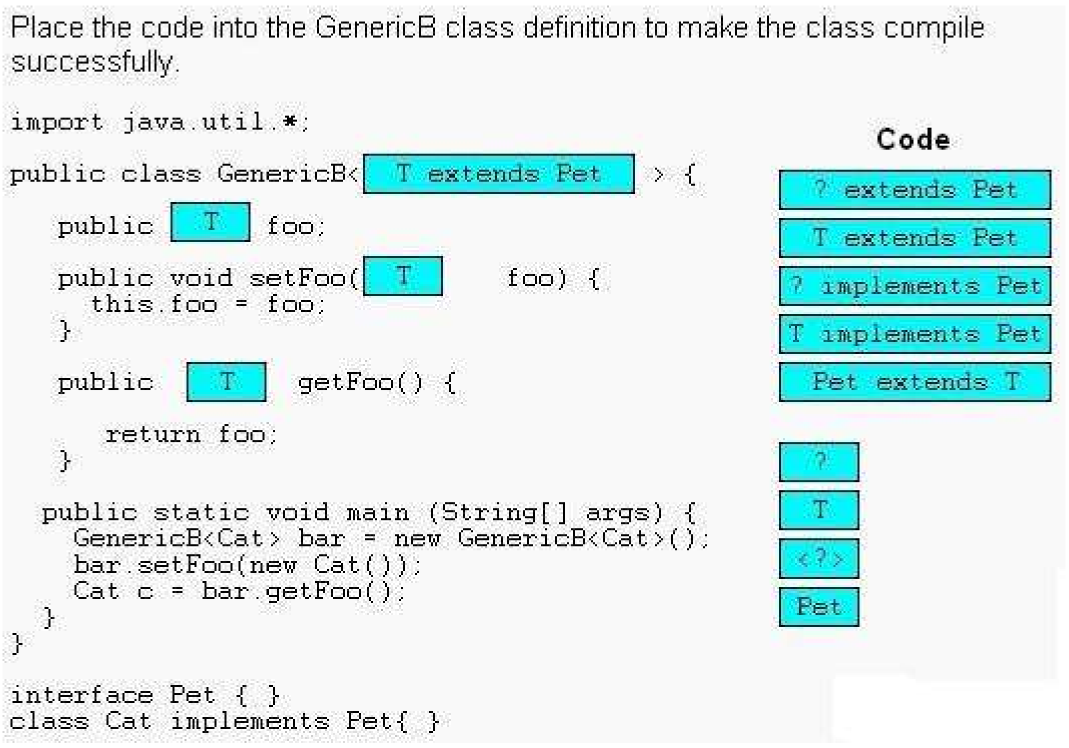
Select and Place:



Place the code into the GenericB class definition to make the class compile successfully.

```
import java.util.*;

public class GenericB< Place > {

   public Place foo;

   public void setFoo( Place   foo) {
     this.foo = foo;
   }

   public  Place  getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

interface Pet { }
class Cat implements Pet{ }
```

Correct Answer:



Place the code into the GenericB class definition to make the class compile successfully.

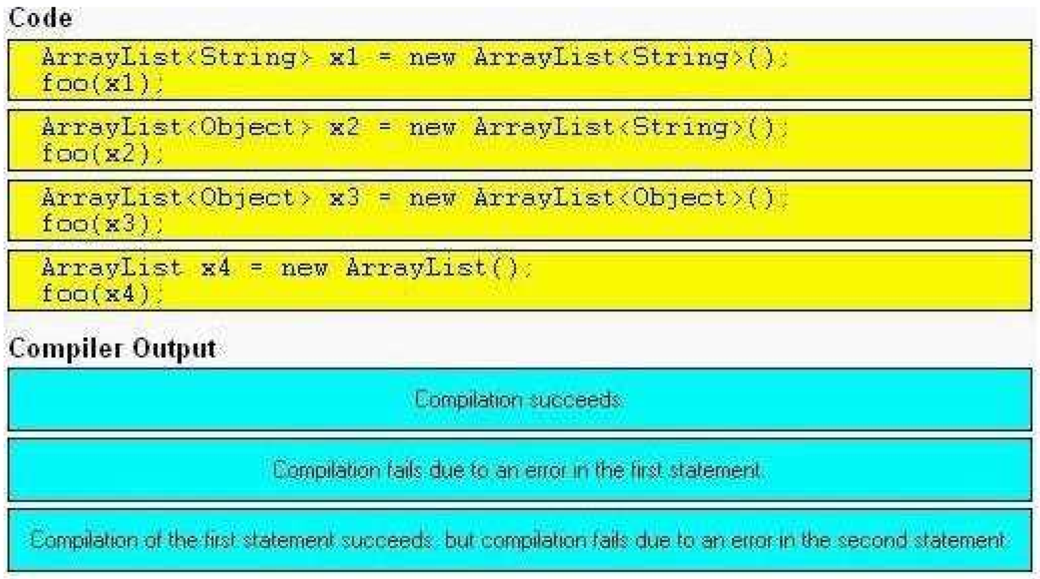
```
import java.util.*;

public class GenericB< T extends Pet > {

   public  T  foo;

   public void setFoo( T    foo) {
     this.foo = foo;
   }

   public  T  getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

interface Pet { }
class Cat implements Pet{ }
```

QUESTION 39

Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
01. import java.util.*;
02. public class X {
03.    public static void main(String[] args) {
04.        // insert code here
05.        // insert code here
06.    }
07.    public static void foo(List<Object> list) { }
08. }
```

Select and Place:



Code

```
ArrayList<String> x1 = new ArrayList<String>();
foo(x1);
```

```
ArrayList<Object> x2 = new ArrayList<String>();
foo(x2);
```

```
ArrayList<Object> x3 = new ArrayList<Object>();
foo(x3);
```

```
ArrayList x4 = new ArrayList();
foo(x4);
```

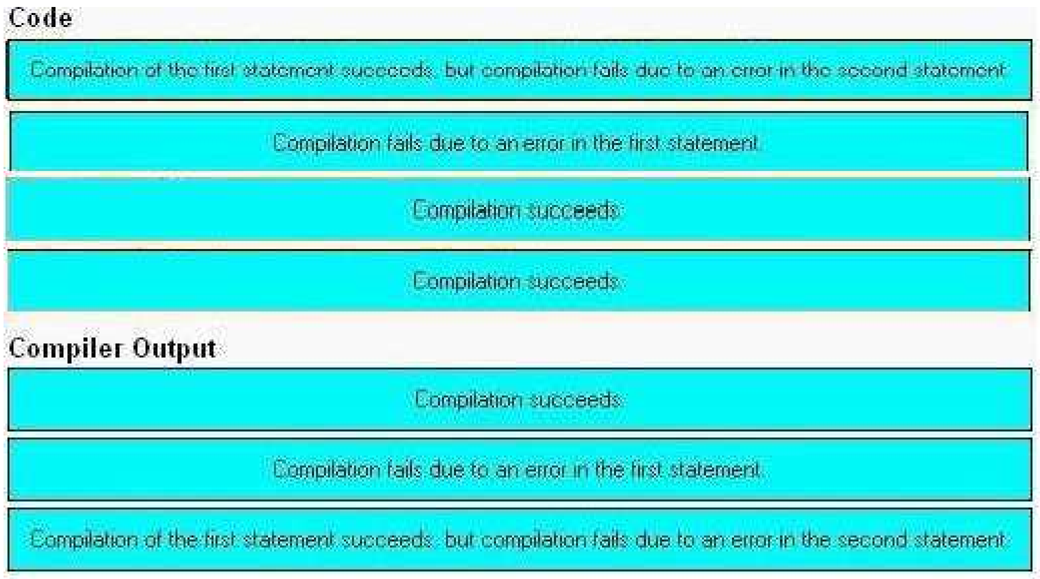
Compiler Output

Compilation succeeds

Compilation fails due to an error in the first statement

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement

Correct Answer:



Code

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement

Compilation fails due to an error in the first statement

Compilation succeeds

Compilation succeeds

Compiler Output

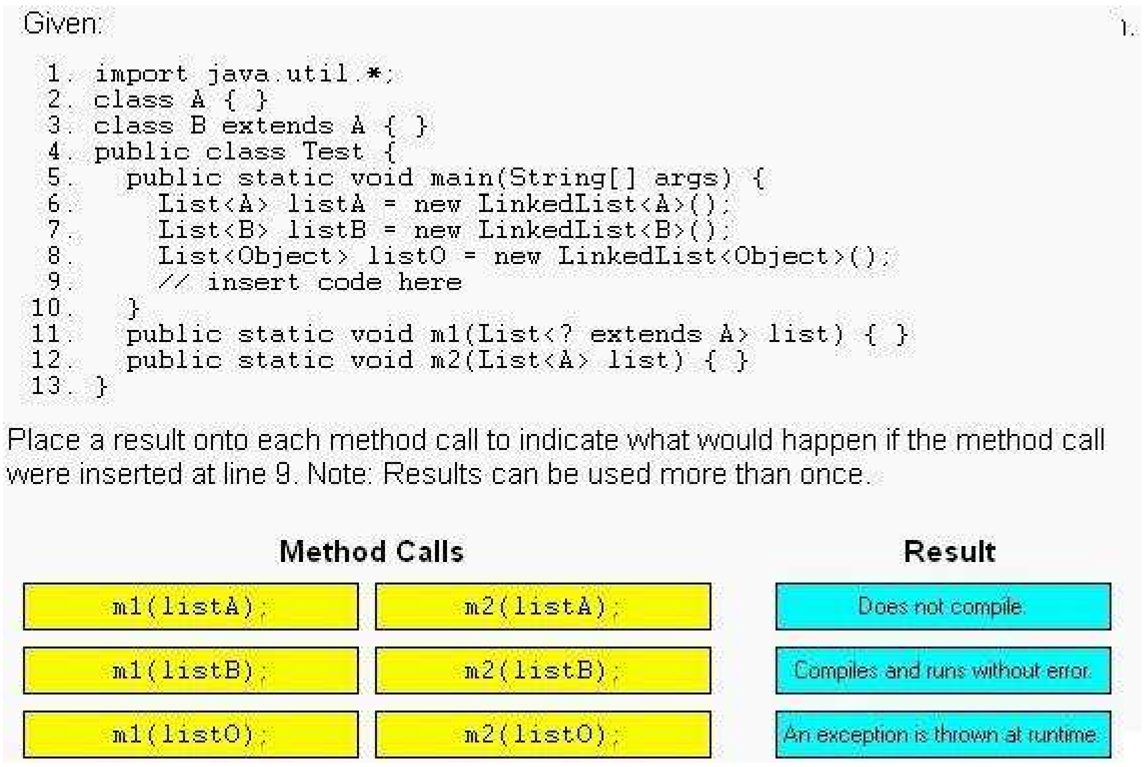
Compilation succeeds

Compilation fails due to an error in the first statement

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement

QUESTION 40

Select and Place:



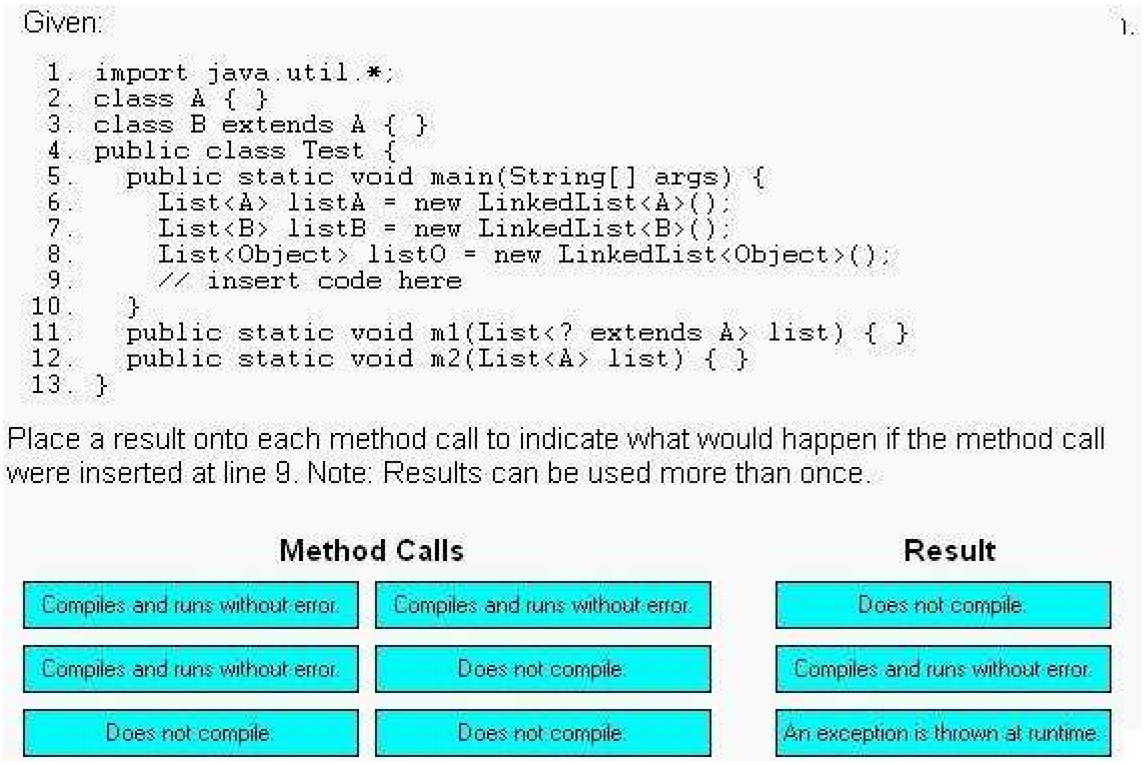
Given:

```
 1. import java.util.*;
 2. class A { }
 3. class B extends A { }
 4. public class Test {
 5.   public static void main(String[] args) {
 6.     List<A> listA = new LinkedList<A>();
 7.     List<B> listB = new LinkedList<B>();
 8.     List<Object> listO = new LinkedList<Object>();
 9.     // insert code here
10.   }
11.   public static void m1(List<? extends A> list) { }
12.   public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
`m1(listA);`	`m2(listA);`	Does not compile.
`m1(listB);`	`m2(listB);`	Compiles and runs without error.
`m1(listO);`	`m2(listO);`	An exception is thrown at runtime.

Correct Answer:



Given:

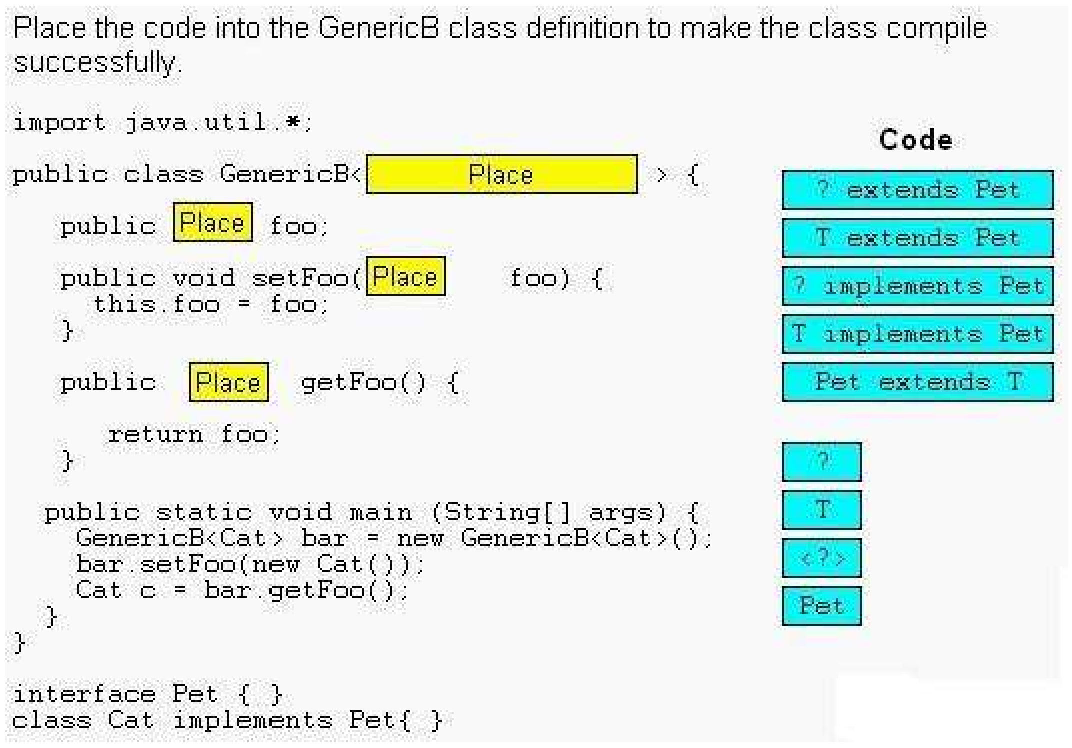
```
 1. import java.util.*;
 2. class A { }
 3. class B extends A { }
 4. public class Test {
 5.   public static void main(String[] args) {
 6.     List<A> listA = new LinkedList<A>();
 7.     List<B> listB = new LinkedList<B>();
 8.     List<Object> listO = new LinkedList<Object>();
 9.     // insert code here
10.   }
11.   public static void m1(List<? extends A> list) { }
12.   public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
Compiles and runs without error.	Compiles and runs without error.	Does not compile.
Compiles and runs without error.	Does not compile.	Compiles and runs without error.
Does not compile.	Does not compile.	An exception is thrown at runtime.

QUESTION 41

Select and Place:



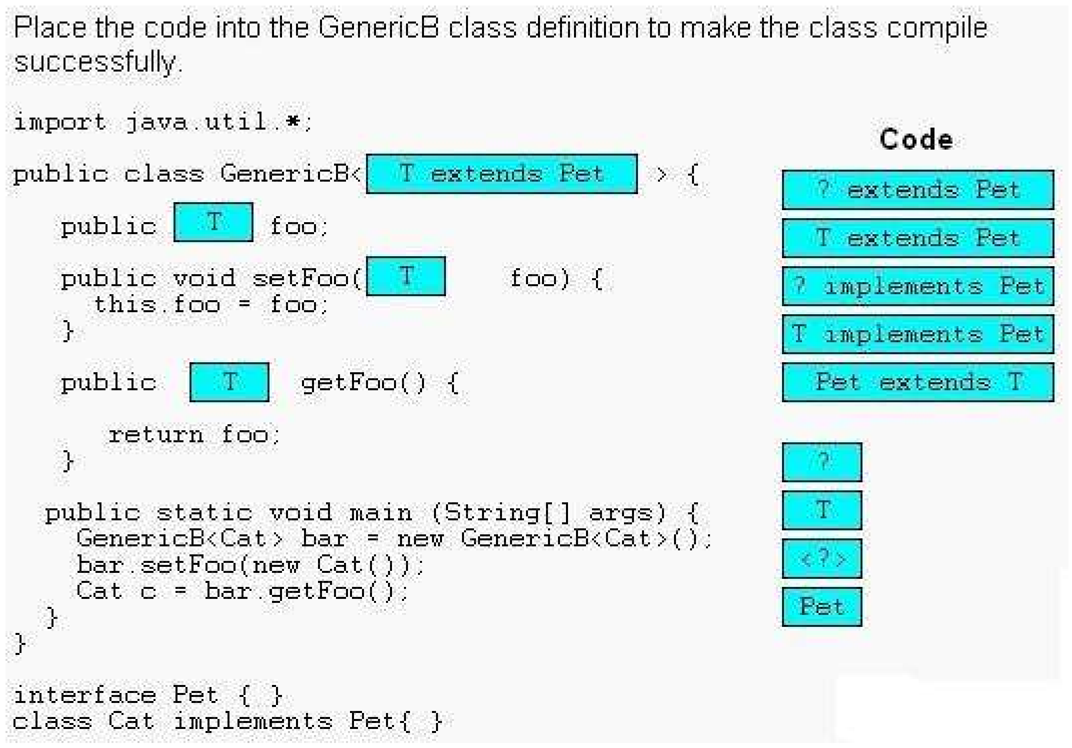
Place the code into the GenericB class definition to make the class compile successfully.

```
import java.util.*;

public class GenericB<     Place     > {

   public Place  foo;

   public void setFoo(Place    foo) {
     this.foo = foo;
   }

   public  Place  getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

interface Pet { }
class Cat implements Pet{ }
```

Code

```
? extends Pet
T extends Pet
? implements Pet
T implements Pet
Pet extends T

?
T
<?>
Pet
```

Correct Answer:



Place the code into the GenericB class definition to make the class compile successfully.

```
import java.util.*;

public class GenericB< T extends Pet > {

   public  T  foo;

   public void setFoo(  T    foo) {
     this.foo = foo;
   }

   public   T   getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

interface Pet { }
class Cat implements Pet{ }
```

Code

```
? extends Pet
T extends Pet
? implements Pet
T implements Pet
Pet extends T

?
T
<?>
Pet
```