

ion 13

et answered
ed out of
ag question

```
public class StringDemo {  
    public static void main(String args[]){  
        StringBuffer c = new StringBuffer("Hello");  
        c.delete(0,2);  
        System.out.println(c);  
    }  
}
```

What is the output of the above program?

Select one:

- a. Hel
- b. None of the answers
- c. llo
- d. He
- e. lo

Next page



NetExam

Sri Lanka Institute of Information Technology

Which of the following exception(s) is(are) will generate a compilation error if not han

Select one or more:

- NumberFormatException
- InterruptedException
- ClassNotFoundException
- IOException
- ArrayIndexOutOfBoundsException



14
answered
out of
question

What is/are the correct sentence(s) from below



Select one or more:

- Like in C++, Java also requires memory allocation and deallocation explicitly by the program code itself
- A Java source file can have multiple classes and only one public class
- Java does not use constructor calls when creating objects
- Java's GC (garbage collector) discards used/unused objects and programmers do not have to manage memory
- Java's new keyword allocates memory for a new object from the Heap and returns a reference to it

acer

F2

F3

F4

F5

F6

F7

F8

F9

F10

#

\$

%

^

&

*

3

4

5

6

Question 14
Not yet answered
Marked out of
1.00
Flag question

What is the output of the given code?

```
public class Test {  
    public static void main(String[] args) {  
        int x = 3;  
        int y = 4;  
        System.out.println(add(x, y));  
        System.out.println(add(x+2, y+x));  
    }  
    public int add(int a, int b) {  
        return a + b;  
    }  
}
```

this should be static in order to run this code
running output = 12

Select one:

- Compilation error
- Runtime error
- None of the given choices



Suppose you need to work with a collection of elements that need to be sorted in their natural ordering, iterated in descending order, and each element has a unique string associated with its value.

Which of the following collections classes in the java.util package best suit your needs for this scenario?

Select one:

- ArrayList
- HashMap
- Vector
- TreeMap
- HashSet

Next

"Persons" too. You are supposed to implement this scenario using Java language and Object oriented concepts. the "Teacher" is inherited from "Employee" class using extends.

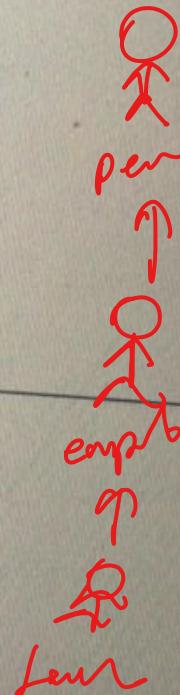
The Object Oriented Concept when a class has two or more extending classes is

Multiple Inheritance

"Teacher" can not be implemented by extending "Person" class too using JAVA.

Hence,

Person should be an interface and Employee should be a class



Nex

What is the result of the following program?

```
1. public abstract class Book {  
2.     public final void read() {  
3.         System.out.println("Reading a Book")  
4.     }  
5.  
6.     public static void main(String [] args)  
7.         Book book = new NonFictionBook();  
8.         book.read();  
9.     }  
10. }  
11.  
12. class NonFictionBook extends Book {  
* 13.     → public void read() {  
14.         System.out.println("Reading a NonFiction  
15.     ")  
16. }
```



Compile error on line 13

Compile error on line 8

Reading a Book

Compile error on line 7



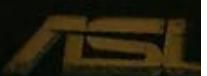
Sri Lanka Institute of Information Technology

What is the most suitable design pattern which describes the following statement?

Encapsulate a request as an object, thereby letting you parametrize clients with different undoable operation

Select one:

- Adapter pattern
- Template pattern
- Strategy pattern
- Bridge pattern
- Command pattern





NetExam

Sri Lanka Institute of Information Technology

Select the wrong statement

Select one:

- All the other statements are correct
- JDK is a part of JRE
- JDK is a part of java SDK
- JVM is a part of JRE
- JVM is a part of JDK

This question is based on Java collections.

A java developer was asked to develop a code where it takes the mathematical expression and check the parenthesis are balanced in that expression.

Write down the class called "Checkparanthesis" where it takes the expression as the user input.

Then check the expression is balanced with the parenthesis and give the relevant output.

Refer the console output below for your reference.

Parenthesis Matching Test

Enter expression

((3+5) *4)-1))+100)

Matches and Mismatches:

```
')) at index 6 matched with ')' at index 2
'(' at index 1 is unmatched
```



NetExam

Sri Lanka Institute of Information Technology

```
public class StringDemo {  
    public static void main(String args[]){  
        StringBuffer c = new StringBuffer("Hello");  
        c.delete(0,2);  
        System.out.println(c);  
    }  
}
```

What is the output of the above program?

Select one:

- a. He
- b. None of the answers
- c. lo
- d. llo
- e. Hel

You are going to implement a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, and palindromes).

You can refer the output given in **Main** class below.

```
public class MyMain {  
  
    public static void main(String[] args) {  
        Collection<Integer> ci = new ArrayList();  
        ci.add(1);  
        ci.add(2);  
        ci.add(3);  
        ci.add(4);  
        int count = Algorithm.countIf(ci, new OddPredicate());  
        System.out.println("Number of odd integers = " + count);  
    }  
}
```

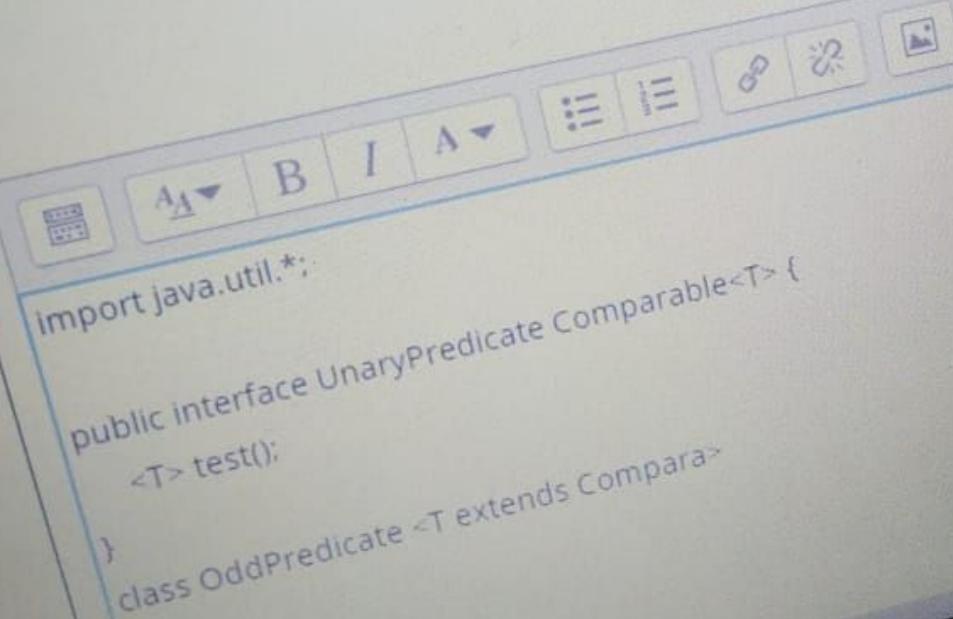
```
num:  
Number of odd integers = 2
```

1. Implement an generic interface called "*UnaryPredicate*" where it have a method call *test* which accepts a generic type and returns if the property existing or not.
2. Implement a class called "*OddPredicate*" where is use the "*UnaryPredicate*" interface.In the *OddPredicate* class, is checks if the given integer is odd or not.
3. Implement the class called "*Algorithm*" where it has a *countIf* method which accept the object and the predicate. It should returns the count which the predicate is existing in the object list.

1. Implement an generic Interface called "Property" which accept the object and the property name.

2. Implement a class called "OddPredicate" where is use the "UnaryPredicate" interface to check if the number is odd or not.

3. Implement the class called "Algorithm" where it has a countIf method which accept the object and the predicate. It will return the count of objects which the predicate is existing in the object list.



The screenshot shows a Java code editor with the following code:

```
import java.util.*;  
  
public interface UnaryPredicate Comparable<T> {  
    <T> test();  
}  
class OddPredicate <T extends Comparable<T> implements UnaryPredicate {
```

The code defines an interface `UnaryPredicate` that extends `Comparable` with a generic type `T`. It contains a single abstract method `test()` that returns a value of type `<T>`. Below it, a class `OddPredicate` is implemented, also generic over `T`, which implements the `UnaryPredicate` interface.

```
int count = Algorithm.countIf(ci, new OddPredicate());  
System.out.println("Number of odd integers = " + count);  
}  
}
```

```
Number of odd integers = 2
```

1. Implement a generic interface called "*UnaryPredicate*" where it has a method call *test* which accepts a generic type and returns a boolean value indicating whether the property exists or not.
2. Implement a class called "*OddPredicate*" where it uses the "*UnaryPredicate*" interface. In the *OddPredicate* class, it checks if the integer is odd or not.
3. Implement the class called "*Algorithm*" where it has a *countIf* method which accepts the object and the predicate. It should return the count of objects for which the predicate is existing in the object list.


```
import java.util.*;  
  
public interface UnaryPredicate <T> {  
  
    public boolean test(T obj);  
  
}  
  
class OddPredicate implements UnaryPredicate<Integer> {  
    public boolean test(Integer i){  
        return i % 2 != 0;  
    }  
  
  
    public final class Algorithm{  
        public static<T> T max (T x,T y){  
            return x > y ? x : y;  
        }  
  
    }  
}
```

```
import java.util.*;  
  
interface UnaryPredicate<T> {  
    public boolean test(T obj);  
}  
  
class OddPredicate implements UnaryPredicate<Integer> {  
    public boolean test (Integer i) {  
        return i % 2 != 0;  
    }  
  
    final class Algorithm {  
        public static <T> int countIf (Collection <T> c, UnaryPredicate<T>  
        int count = 0;  
        for (T elem : c)  
            if (p.test(elem))  
                ++count;  
        return count;  
    }  
}
```

19
20
21

import java.util.*;

public class DisplayWhetherCondition extends Thread{
 this.t = t;
}

public void run(){
 synchronized(t){
 System.out.println("Next day started.....");
 }
}

public class SunnyDay extends Thread(

MyMain t:

public SunnyDay(MyMain t) {
 this.t = t;
}

public void run(){
 synchronized(t){
 System.out.println("It is a sunny day");
 }
}



Batik printing company is printing their design using computer program and which is drawn using two concurrent Threads. Customer allowed to enter pattern styles through keyboard inputs and customer should select number of occurrences (count) to be printed the style. Each thread should print patterns one after the other and you should print the triangle shape using given style.

Refer the console output and implement the two threads in the given space below.

Output:

```
Enter Pattern 1 = +
Enter Pattern 2 = -
Enter count = 6
=====Threads start printing patterns.=====

```

```
+
+
+
+
+
-
-
-
-
-
+
+
+
+
+
-
-
-
-
-
+
+
+
+
+
-
-
-
-
-
+
+
+
+
+
-
```

```
public void run(){
    synchronized(t){
        System.out.println("Next day started.....");
    }
}

public class SunnyDay extends Thread{
    Mymain t;

    public SunnyDay(Mymain t) {
        this.t = t;
    }

    public void run(){
        synchronized(t){
            System.out.println("It is a Sunny day");
            System.out.println("Ending Sunny day");
        }
    }
}
```

```
public static void main(String[] args) {  
    Helper.getChoise("plant").getPlant("Flowering");  
    Helper.getChoise("plant").getPlant("land").bu...  
    Helper.getChoise("soil").addSoil("clay").addS...  
}
```

1. Abstract Factory Pattern

2.

```
public interface PlantType{  
    public void getPlant();  
}  
  
public interface SoilType{  
    public void addSoil();  
}  
  
public class Flowering implements PlantType{  
    public void getPlant(){  
        System.out.println("Buy FloweringPlant");  
    }  
}  
  
public class LandPlant implements PlantType{  
    public void getPlant(){  
        System.out.println("Buy LandPlant");  
    }  
}
```

```
public static void main(String[] args) {  
    Helper.getChoise("plant").getPlant("Flowering");  
    Helper.getChoise("plant").getPlant("land").bu...  
    Helper.getChoise("soil").addSoil("clay").addS...  
}
```

1. Abstract Factory Pattern

2.

```
public interface PlantType{  
    public void getPlant();  
}  
  
public interface SoilType{  
    public void addSoil();  
}  
  
public class Flowering implements PlantType{  
    public void getPlant(){  
        System.out.println("Buy FloweringPlant");  
    }  
}  
  
public class LandPlant implements PlantType{  
    public void getPlant(){  
        System.out.println("Buy LandPlant");  
    }  
}
```



A▼

B

I

A▼



1. Template Design Pattern

2.

```
public abstract class Orientation{
```

```
    public final void papers(){
```

```
        MathsPaper();
```

```
        SoftSkillsPaper();
```

```
        SpecialPaper();
```

```
}
```

```
    public abstract void SpecialPaper();
```

```
    public void MathsPaper(){
```

```
        System.out.println("Mathematics Paper");
```

```
}
```

```
    public void SoftSkillsPaper(){
```

```
        System.out.println("Softskills Paper");
```

```
}
```

```
}
```

```
public class SoftwareEngineering extends Orientation{
```

```
    public void SpecialPaper(){
```

```
        System.out.println("Object Oriented Programming Paper");
```

```
}
```

```
}
```

```
public class DataScience extends Orientation{
```

```
    public void SpecialPaper(){
```

```
        System.out.println("Machine Learning Paper");
```

Question 18

Not yet answered

Marked out of
10.00

Flag question

This question is based on Design patterns.

Assume that each Information Technology student needs to pass mathematics and also demonstrate soft skills (such as Critical Thinking, Team work and so on) in their Orientation to obtain their degrees. Coming Semesters, course will add some special papers to their courses based on their chosen paths (programming or Database).

1. Write the most suitable design pattern name that can be used in above scenario?
2. Implement the relevant classes to demonstrate the design pattern you named in part 1.

You can refer to the main class given below and implement the classes accordingly.

```
1
2 public class Demo {
3
4     public static void main(String args[]) {
5         Orientation ob1= new SoftwareEngineering();
6         System.out.println("Software Engineering paper");
7         ob1.papers();
8         ob1=new DataScience();
9         System.out.println("\nData Science paper");
10        ob1.papers();
11    }
12 }
13
```

```
<terminated> Demo () [Java Application] C:\Program Files\Java\jre1.8.0
Software Engineering paper
Mathematics paper
Softskills paper
Object Oriented Programming paper
Data Science paper
Mathematics paper
Softskills paper
Machine Learning paper
```

```
public abstract class Orientation{
    public final void papers(){
        MathsPaper();
        SoftSkillsPaper();
        SpecialPaper();
    }
    public abstract void SpecialPaper();
    public void MathsPaper(){
        System.out.println("Mathematics Paper");
    }
    public void SoftSkillsPaper(){
        System.out.println("Softskills Paper");
    }
}

public class SoftwareEngineering extends Orientation{
    public void SpecialPaper(){
        System.out.println("Object Oriented Programming Paper");
    }
}

public class DataScience extends Orientation{
    public void SpecialPaper(){
        System.out.println("Machine Learning Paper");
    }
}
```

2:

```
public interface PlantType {  
    public void getPlant();  
}  
  
public interface SoliType {  
    public void addSoli();  
}  
  
public class Flowering implements PlantType {  
    public void getPlant() {  
        System.out.println("Buy Flowering Plant");  
    }  
}  
  
public class LandPlant implements PlantType {  
    public void getPlant() {  
        System.out.println("Buy LandPlant");  
    }  
}  
  
public class Flowering implements SoliType{  
    public void addSoli() {  
        System.out.println("Bought Soli");  
    }  
}  
  
public class LandPlant implements SoliType{  
    public void addSoli() {
```

```
public class LandPlant implements PlantType {  
    public void getPlant() {  
        System.out.println("But LandPlant");  
    }  
}
```

```
public class Flowering implements SoliType{  
    public void addSoli() {  
        System.out.println("Bought Soli");  
    }  
}
```

```
I  
public class LandPlant implements SoliType{  
    public void addSoli() {  
        System.out.println("Bought Clay");|  
    }  
}
```

Question 18

Not yet answered

Marked out of
10.00

Flag question

This question is based on Design patterns.

Assume that each Information Technology student needs to pass mathematics and also demonstrate soft skills (such as Critical Thinking, Team work and so on) in their Orientation to obtain their degrees. Coming Semesters, course will add some special papers to their courses based on their chosen paths (programming or Database).

1. Write the most suitable design pattern name that can be used in above scenario?
2. Implement the relevant classes to demonstrate the design pattern you named in part 1.

You can refer to the main class given below and implement the classes accordingly.

```
1
2 public class Demo {
3
4     public static void main(String args[]) {
5         Orientation ob1= new SoftwareEngineering();
6         System.out.println("Software Engineering paper");
7         ob1.papers();
8         ob1=new DataScience();
9         System.out.println("\nData Science paper");
10        ob1.papers();
11    }
12 }
13
```

```
<terminated> Demo () [Java Application] C:\Program Files\Java\jre1.8.0
Software Engineering paper
Mathematics paper
Softskills paper
Object Oriented Programming paper
Data Science paper
Mathematics paper
Softskills paper
Machine Learning paper
```

Marked out of
1.00

Flag question

```
public static void main(String[] args) {
    S2 s2 = new S2();
    System.out.println(s2.add(2, 3));
    System.out.println(s2.add("abc", "def"));
    S1 s1 = new S1();
    System.out.println(s1.add(3, 5));
    System.out.println(s1.add("my ", "pet"));
}

abstract class S1 {

    abstract int add(int a, int b);

    public String add(String a, String b) {
        return a + "," + b;
    }
}

class S2 extends S1 {

    public int add(int a, int b) {
        return a + b;
    }
}
```

Select one or more:

- S1 s1 = new S1(); is wrong as we cannot create an object from Class S1
- S2 s2 = new S2(); is wrong as we cannot create an object from class S2
- S1 should not be defined as an abstract class as String add(String a, String b) has an implementation
- Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong
- Cannot use s2.add("abc", "def") as that method is not implemented in class S2

Finish attempt ...

Time left 1:57:10



MCQ SECTION

1	2	3	4
8	9	10	11
15			

ESSAY SECTION

16	17	18

FEEDBACK QUESTION

19



Question 1

Not yet answered

Marked out of
1.00 Flag question

What is true about java constants in coding?

Select one or more:

- When declaring, all the letters in the constants should be upper case X
- Constants can remove magic numbers in the code ✓
- You cannot change the value in the constant ✓
- Constant will be inherited to sub classes X
- A global constant should be default and static ✓

```
public static void main(String[] args) {
    S2 s2 = new S2();
    System.out.println(s2.add(2, 3));
    System.out.println(s2.add("abc", "def"));
    S1 s1 = new S1();
    System.out.println(s1.add(3, 5));
    System.out.println(s1.add("my ", "pet"));
}

abstract class S1 {

    abstract int add(int a, int b);

    public String add(String a, String b) {
        return a + "," + b;
    }
}

class S2 extends S1 {

    public int add(int a, int b) {
        return a + b;
    }
}
```

Select one or more:

- S1 s1 = new S1(); is wrong as we cannot create an object from Class S1
- S2 s2 = new S2(); is wrong as we cannot create an object from class S2
- S1 should not be defined as an abstract class as String add(String a, String b) has an implementation
- Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong
- Cannot use s2.add("abc", "def") as that method is not implemented in class S2

Finish attempt ...

Time left 1:57:10



MCQ SECTION

1	2	3	4
8	9	10	11
15			

ESSAY SECTION

16	17	18

FEEDBACK QUESTION

19

NetExam

Sri Lanka Institute of Information Technology

What is the result of the following program?

```
1. public abstract class Book {  
2.     public final void read() {  
3.         System.out.println("Reading a Book");  
4.     }  
5.  
6.     public static void main(String [] args) {  
7.         Book book = new NonFictionBook();  
8.         book.read();  
9.     }  
10. }  
11.  
12. class NonFictionBook extends Book {  
13.     public void read() {  
14.         System.out.println("Reading a NonFictionBook");  
15.     }  
16. }
```

- Compile error on line 7
- Reading a NonFictionBook
- Compile error on line 13
- Compile error on line 8

Select one

Reading a NonFictionBook



acer



Question 1

Not yet answered

Marked out of
1.00

Flag question

What is the output of the following code?

```
class ABC {  
    static int a = 5;  
    static {  
        a = a*3;  
    }  
}  
  
public class Test {  
  
    public static void main(String[] args) {  
        System.out.println(ABC.a);  
    }  
}
```

Select one:

- Runtime error
- None of the given choices
- Compilation error
- 5
- 15



Question 2

Not yet answered

Marked out of
1.00

Flag question

When an exception is **throws** , the **Exception Handler** searches the try statement's catch clauses from top to bottom and passes control of the program to the **first matching** catch exception.

thrown **catch**

JVM

system

exact

super class



Next page

NetExam

Sri Lanka Institute of Information Technology

Which of the following exception(s) is(are) will generate a compilation error if not handled?

Select one or more:

- `ArrayIndexOutOfBoundsException`
- `ClassNotFoundException`
- `NumberFormatException`
- `InterruptedException`
- `IOException`

acer

F3

F4

F5

F6

F7

F8

F9

Question 1

Not yet answered

Marked out of
1.00

 Flag question

which of these packages String and StringBuffer classes are implemented?

Select one:

- a. java.lang
- b. None of the given
- c. java.String
- d. java.util
- e. java.awt



Next page

The output of the following fraction of code is

```
public class Tarzan {  
    public static void main(String args[]) {  
        String s1 = "Jane";  
        String s2 = "Jane";  
        System.out.println(s1 == s2);  
    }  
}
```



Select one:

- a. Throws an Exception
- b. JaneJane
- c. false
- d. true
- e. jane



Choose... ▾

Choose...

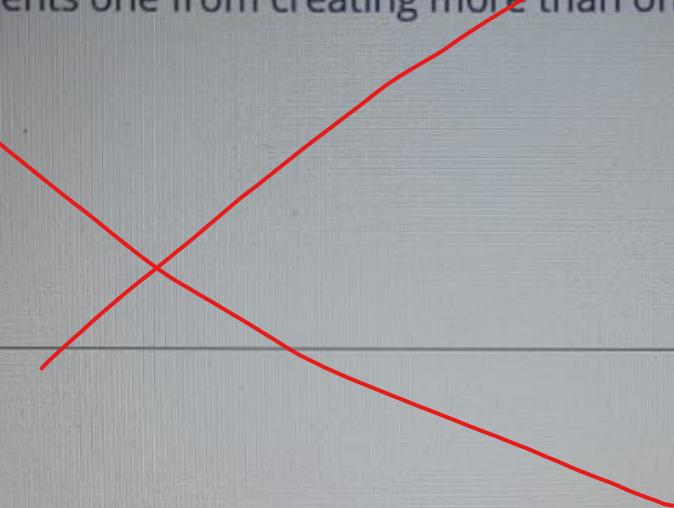
Singleton

Factory

Facade

Observer

pattern prevents one from creating more than one instance of a variable.



What is the output of given code?

```
public class Test {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder("I love java");  
        ?.  
        sb.substring(0,9);  
        sb.reverse();  
        System.out.println(sb);  
    }  
}
```

Select one:

- avaj evol I
- I love jav
- vaj evol I
- I love Java
- aj evol I



Question 2

Not yet answered

Marked out of
1.00

Flag question

In which of the following pattern, a class behavior or its algorithm can be changed?

Select one:

- Command pattern
- Abstract factory pattern
- Template pattern
- Strategy pattern
- Simple factory pattern



What is the output of the following code?

```
class ABC {  
    final int a = 20;  
  
    public ABC(int val) {  
        a = val;  
    }  
}  
  
public class Test {  
  
    public static void main(String[] args) {  
  
        ABC abc = new ABC(30);  
        System.out.println(abc.a);  
    }  
}
```

Select one:

- 30
- Non of the given choices
- 20
- Runtime error
- Compilation error



What is the output of the following code

```
public class Test {  
    public static void main(String[] args) {  
        int height = 5;  
        int base = 3;  
        double area = 1/2 * height * base;  
        System.out.println("Area is : " + area);  
    }  
}
```

8.0 0.0 7.0 7.5 0

Select one:

- Area is : 8.0
- Area is : 0.0
- Area is : 7.0
- Area is : 7.5
- Area is : 0



NetExam

Sri Lanka Institute of Information Technology

Suppose you need to work with a collection of elements that need to be sorted. Each element has a unique string associated with its value.

Which of the following collections classes in the java.util package best suits this requirement?

Select one:

- TreeMap
- HashSet
- Vector
- HashMap
- ArrayList

ation 3

yet answered
ed out of
ag question

Given the following class definitions

```
1 public class Parent {  
2     protected void sayHi() {  
3         System.out.print("Hi");  
4     }  
5 }  
6  
7 class Child extends Parent {  
8     public void sayHi() {  
9         System.out.print("Hello");  
10    }  
11 }
```

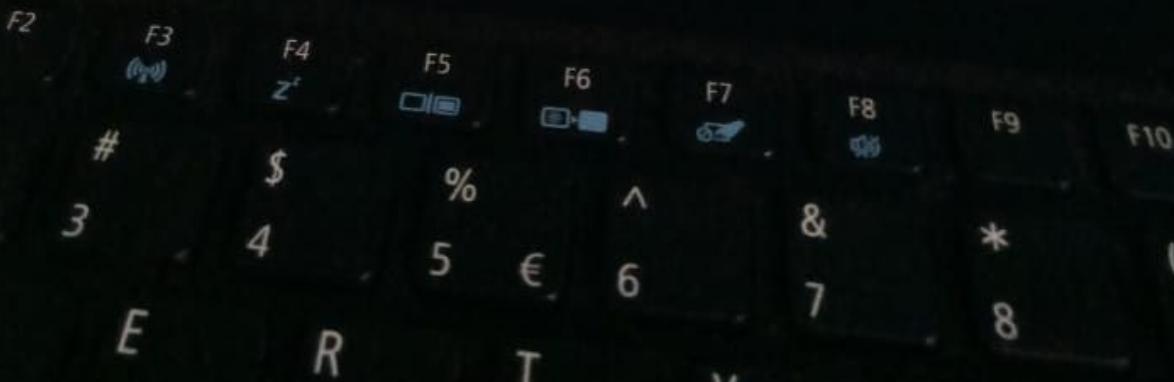
Parent p = new Child();
p.sayHi();

Select one or more:

- Line 16 causes an exception to be thrown.
- Compiler error on line 8
- Hello
- Hi
- Compiler error on line 15

?

acer



Which of the following statements are true?

Select one or more:

- The StringBuilder and StringBuffer classes define the exact same public methods.
- All string literals are automatically instantiated into a String object.
- A StringBuffer object can increase its length when appending characters.
- In a multithreaded environment, use StringBuilder instead of StringBuffer .
- StringBuilder object is immutable.

```
public static void main(String[] args) {
    S2 s2 = new S2();
    System.out.println(s2.add(2, 3));
    System.out.println(s2.add("abc", "def"));
    S1 s1 = new S1();
    System.out.println(s1.add(3, 5));
    System.out.println(s1.add("my ", "pet"));
}

abstract class S1 {

    abstract int add(int a, int b);

    public String add(String a, String b) {
        return a + "," + b;
    }
}

class S2 extends S1 {

    public int add(int a, int b) {
        return a + b;
    }
}
```

Select one or more:

- ~~S1 s1 = new S1();~~ is wrong as we cannot create an object from Class S1
- ~~S2 s2 = new S2();~~ is wrong as we cannot create an object from class S2
- ~~S1 should not be defined as an abstract class as String add(String a, String b) has an implementation~~
- ~~Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong~~
- ~~Cannot use s2.add("abc", "def") as that method is not implemented in class S2~~

Finish attempt ...

Time left 1:57:10



MCQ SECTION

1	2	3	4
8	9	10	11
15			

ESSAY SECTION

16	17	18
----	----	----

FEEDBACK QUESTION

19



NetExam

Sri Lanka Institute of Information Technology

Question 1

1 yet answered
1 ed out of

8 Question

Which of these class is used to create an object whose character sequence is changeable?

Select one:

- a. None of the mentioned
- b. String
- c. StringBuffer
- d. Both of the mentioned



NetExam

Sri Lanka Institute of Information Technology

Question 1

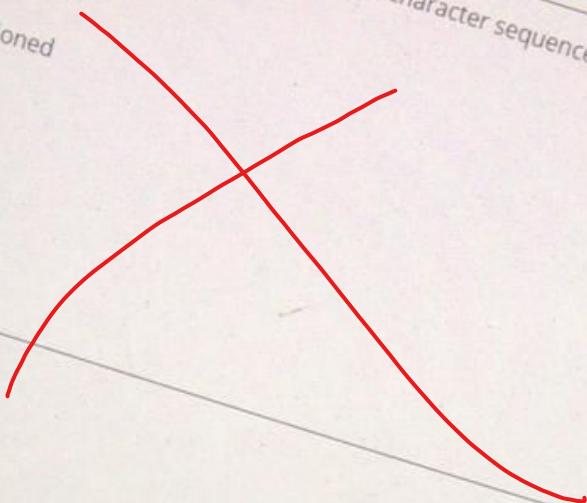
1 yet answered
1 out of

8 question

Which of these class is used to create an object whose character sequence is changeable?

Select one:

- a. None of the mentioned
- b.
- c. StringBuffer
- d. Both of the mentioned



```
file x
question

3     static int no = 10;
4
5
6* static{
7    no = no + add(no);
8    System.out.println(no); -----
9 }
10 -----
11* static int add(int no){
12    return no;
13 }
14 -----
15* {
16    System.out.println(no);
17 }
18 -----
19* public Ex01() {
20    System.out.println("Construcor");
21 }
22 -----
23* public static void main(String[] args) {
24    Ex01 ex01 = new Ex01();
25 }
26 }
```

Select one:

20

20

Constructor

Compile Error

Constructor

10





NetExam

Sri Lanka Institute of Information Technology

Question 1

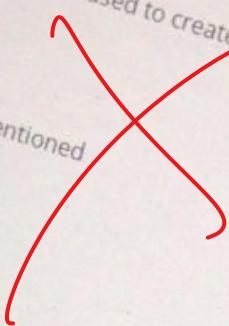
1 yet answered
1 ed out of

8 Question

Which of these class is used to create an object whose character sequence is changeable?

Select one:

- a. None of the mentioned
- b. String
- c. StringBuffer
- d. Both of the mentioned



Not yet answered

Marked out of
1.00

Flag question

What is / are the error /s in the following program?

```
class Main {  
    public static void main(String[] args) {  
        S2 s2 = new S2();  
        System.out.println(s2.add(2, 3));  
        System.out.println(s2.add("abc", "def"));  
        S1 s1 = new S1();  
        System.out.println(s1.add(3, 5));  
        System.out.println(s1.add("my ", "pet"));  
    }  
  
    abstract class S1 {  
        abstract int add(int a, int b);  
        public String add(String a, String b) {  
            return a + "," + b;  
        }  
    }  
  
    class S2 extends S1 {  
        public int add(int a, int b) {  
            return a + b;  
        }  
    }  
}
```



Select one or more:

- S1 should not be defined as an abstract class as String add(String a, String b) has an implementation
- Cannot use s2.add("abc", "def") as that method is not implemented in class S2
- S1 s1 = new S1(); is wrong as we cannot create an object from Class S1
- S2 s2 = new S2(); is wrong as we cannot create an object from class S2
- Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong



What is the output of the following code ?

```
public class Test {  
  
    public static void main(String[] args) {  
        int i = 5;  
        while (i<10) {  
            System.out.println(i++);  
            System.out.println(++i);  
        }  
    }  
}
```

i = 567891011

5
6
7
8
9
10
11

Select one:

- 5
- 8
- 11
- 16
- 6

≡ Quiz navigation

Finish attempt ...

Time left 1:57:04



MCQ SECTION

1	2	3	4
8	9	10	11
15			

ESSAY SECTION

16	17	18
19		

FEEDBACK QUESTION

10
○
5
6
6
7
7
8
8
9
9
10



5
7
7
9
9
11

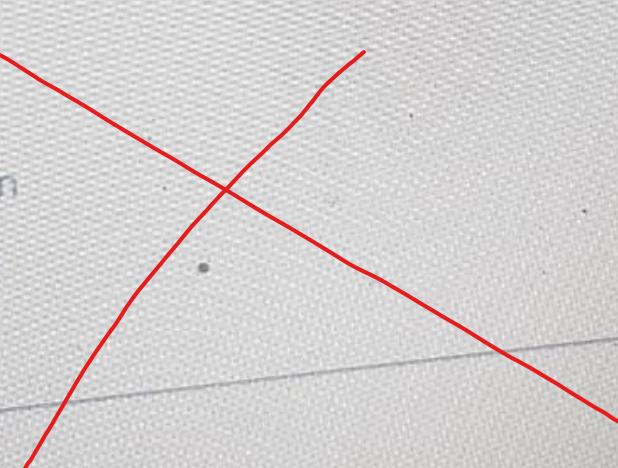


Infinite loop

In which of the following pattern, a class behavior or its algorithm can be changed at run time?

Select one:

- Strategy pattern
- Command pattern
- Abstract factory pattern
- Simple factory pattern
- Template pattern



Given the following class definition:

```
public class PrintA extends Thread {  
    public void run() {  
        System.out.print("A");  
    }  
}
```

which of the statement is true about the following program?

```
public class PrintB {  
    public static void main(String [] args) {  
        Thread a = new PrintA();  
        a.run();  
        System.out.print("B");  
    }  
}
```

Select one:

- The program generates an exception at runtime
- The program does not compile
- The output is always B
- The output is always AB
- The output varies and is either AB or BA

≡ Quiz

Finish atten

Time left 1:



MCQ SECT



ESSAY SEC



FEEDBACK



idle | X | i | +
cursor
question

```
3     static int no = 10;
4
5
6*   static{
7       no = no + add(no);
8       System.out.println(no);
9   }
10
11*  static int add(int no){
12      return no;
13  }
14
15*
16      System.out.println(no);
17  }
18
19* public Ex01(){
20     System.out.println("Construcor");
21 }
22
23* public static void main(String[] args) {
24     Ex01 ex01 = new Ex01();
25 }
26 }
```

Select one:

- 20
- 20
- Constructor
- Compile Error
- Constructor
- 10



? * % ^ & * 1 2 3 4 5 6 7 8 9
W E R T Y U V

The String method compareTo() returns

Select one:

- a. false
- b. -1
- c. +1
- d. true
- e. an Integer value

Question 1

Not yet answered

Marked out of
1.00

 Flag question

which of these packages String and StringBuffer classes are implemented?

Select one:

- a. java.lang
- b. None of the given
- c. java.String
- d. java.util
- e. java.awt



Next page

NetExam

Sri Lanka Institute of Information Technology

What is the result of the following program?

```
1. public abstract class Book {  
2.     public final void read() {  
3.         System.out.println("Reading a Book");  
4.     }  
5.  
6.     public static void main(String [] args) {  
7.         Book book = new NonFictionBook();  
8.         book.read();  
9.     }  
10. }  
11.  
12. class NonFictionBook extends Book {  
13.     public void read() {  
14.         System.out.println("Reading a NonFictionBook");  
15.     }  
16. }
```

- Compile error on line 7
- Reading a NonFictionBook
- Compile error on line 13
- Compile error on line 8

Select one

Reading a NonFictionBook



acer

What is the output of the following code?

```
class ABC {  
    static int a = 5;  
    static {  
        a = a*3;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        System.out.println(ABC.a);  
    }  
}
```

Select one:

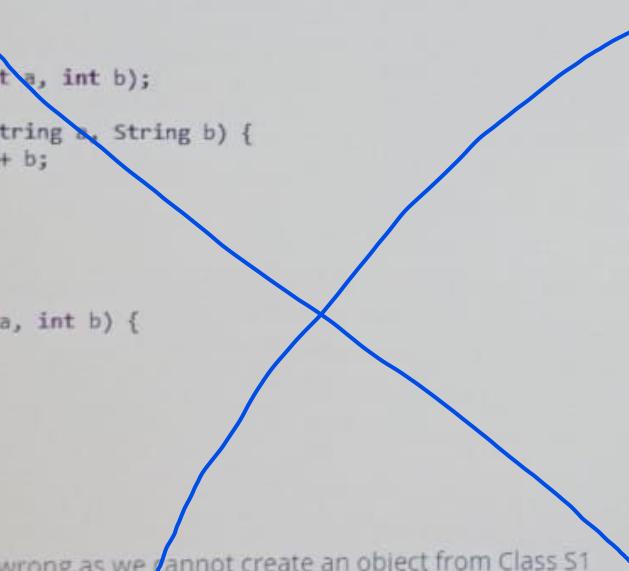
- Compilation error
- 15
- 5
- Runtime error
-

None of the given choices

```
public static void main(String[] args) {
    S2 s2 = new S2();
    System.out.println(s2.add(2, 3));
    System.out.println(s2.add("abc", "def"));
    S1 s1 = new S1();
    System.out.println(s1.add(3, 5));
    System.out.println(s1.add("my ", "pet"));
}

abstract class S1 {
    abstract int add(int a, int b);
    public String add(String a, String b) {
        return a + "," + b;
    }
}

class S2 extends S1 {
    public int add(int a, int b) {
        return a + b;
    }
}
```



Select one or more:

- S1 s1 = new S1(); is wrong as we cannot create an object from Class S1
- S2 s2 = new S2(); is wrong as we cannot create an object from class S2
- S1 should not be defined as an abstract class as String add(String a, String b) has an implementation
- Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong
- Cannot use s2.add("abc", "def") as that method is not implemented in class S2

Finish attempt ...

Time left 1:57:10



MCQ SECTION

1	2	3	4
8	9	10	11
15			

ESSAY SECTION

16	17	18
----	----	----

FEEDBACK QUESTION

19

The String method compareTo() returns

Select one:

- a. false
- b. -1
- c. +1
- d. true
- e. an Integer value

What is/are the correct sentence(s) from below

Select one or more:

- Java does not uses constructor calls when creating objects
- Java's GC (garbage collector) discards used/unused objects and programmers do not have to worry in deallocated memory
- Java's new keyword allocates memory for a new object from the Heap and returns a reference to that memory area
- Like in C++, Java also requires memory allocation and deallocation explicitly by the programmer within the program code itself
- A Java source file can have multiple classes and only one public class



NetExam

Sri Lanka Institute of Information Technology

What is the output of this program?

```
class Output {
    public static void main(String args[])
    {
        String c = ("Hello, I love java programming").substring(3, 9);
        System.out.println(Character.isLowerCase(c.charAt(2)));
    }
}
```

Select one:

- a. 1
- b. A StringOutOfBoundsException will be thrown
- c. true
- d. false
- e. Compilation Error

NetExam

Sri Lanka Institute of Information Technology

Select the TRUE sentence(s) from the below

this()

Select one or more:

- Constructors are declared as void methods as those do not return any value
- this and this() are used for different purposes in Java
- super.super() calls the default constructor of the parent class
- this() calls the default constructor of the current instance
- Constructor can take only one argument





Question 1

Not yet answered

Marked out of
1.00

Flag question

What is true about java constants in coding?

Select one or more:

- When declaring, all the letters in the constants should be upper case
- Constants can remove magic numbers in the code
- You cannot change the value in the constant
- Constant will be inherited to sub classes
- A global constant should be default and static

Moodle

→ X C O D

```
public static void main(String[] args) {  
    int opt = 5;  
    switch (opt){  
        case 1:  
            System.out.println("*****");  
        case 3:  
            System.out.println("$$$$$");  
            break;  
        case 5:  
            System.out.println("#####");  
        default:  
            System.out.println("&&&&&");  
    }  
}
```

(ft)
8 8 0 2 1

Select one:

- &&&&
- Compilation error
- #####
- #####
- Run time exception

acer



Question 2

Not yet answered

Marked out of
1.00

Flag question

When an exception is **throws**, the **Exception Handler** searches the try statement's catch clauses from top to bottom and passes control of the program to the **first matching** catch exception.

thrown **catch**

JVM

system

exact

super class

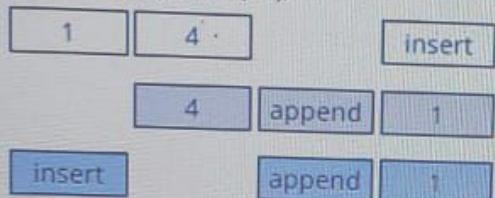
Next page

NetExam

Sri Lanka Institute of Information Technology

Complete the following code so that it compile successfully and prints the output as *abbaccca*

```
StringBuilder sb = new StringBuilder();
sb.append("aaa").insert(1, "bb").insert(4, "ccc");
System.out.println(sb);
```



aa9

abbaa
abbaccca

ASUS

Select the TRUE sentence(s) from the below

Select one or more:

- this and this() are used for different purposes in Java
- this() calls the default constructor of the current instance
- super.super() calls the default constructor of the parent class
- Constructors are declared as void methods as those do not return any value
- Constructor can take only one argument

Next page

If your code does not handle an exception when it is thrown, it is dealt with by default exception handler .

the operating system

system debugger

default exception generator

Java Scheduler

Given the following class definition:

```
public class PrintA extends Thread {  
    public void run() {  
        System.out.print("A");  
    }  
}
```

which of the statement is true about the following program?

```
public class PrintB {  
    public static void main(String [] args) {  
        Thread a = new PrintA();  
        a.run();  
        System.out.print("B");  
    }  
}
```

Select one:

- The program generates an exception at runtime
- The program does not compile
- The output is always AB
- The output varies and is either AB or BA
- The output is always B

What is the output of the following code?

```
class ABC {  
    final int a = 20;  
  
    public ABC(int val) {  
        a = val;  
    }  
}  
  
public class Test {  
  
    public static void main(String[] args) {  
  
        ABC abc = new ABC(30);  
        System.out.println(abc.a);  
    }  
}
```

Select one:

- 30
- Runtime error
- Compilation error
- None of the given choices
- 20



on 8

answered

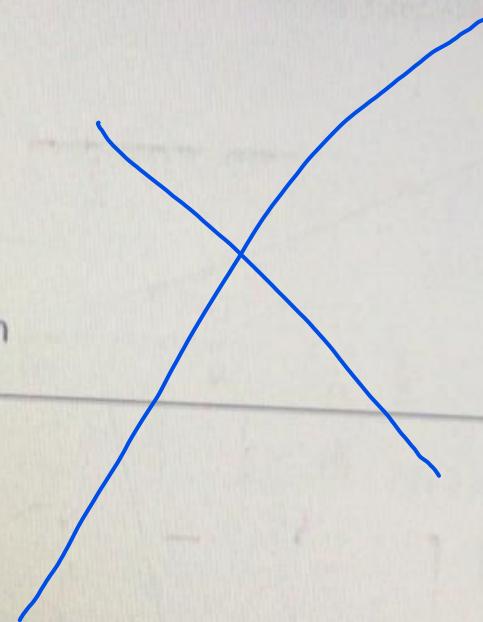
out of

g question

Which design pattern works on data and action have taken based on data provided?

Select one:

- Template
- Facade
- Singleton
- Strategy pattern
- Command Pattern



Which of the following is wrapper class for data type char?

Select one:

- a. Char
- b. Letter
- c. Character
- d. Float
- e. String

Net Exam

Sri Lanka Institute of Information Technology

Which of the following statements are true?

Select one or more:

- The StringBuilder and StringBuffer classes define the exact same public methods
- A StringBuffer object can increase its length when appending characters
- All string literals are automatically instantiated into a String object.
- In a multithreaded environment, use StringBuilder instead of StringBuffer
- A StringBuilder object is immutable

acer

F3

(F3)

F4

(F4)

F5

(F5)

F6

(F6)

F7

(F7)

Given the following class definition:

1. **public abstract class MyWindowCloser {**
2. **protected abstract void closeWindow(String id);**
3. **}**

which of the following methods could appear in a child class of MyWindowCloser ?

Select one or more:

- protected void closeWindow(String id)**
- void closeWindow(String id)**
- protected int closeWindow(String id)**
- public int closeWindow(String x)**
- private void closeWindow()**

Given the following class definition:

```
public class PrintA extends Thread {  
    public void run() {  
        System.out.print("A");  
    }  
}
```

which of the statement is true about the following program?

```
public class PrintB {  
    public static void main(String [] args) {  
        Thread a = new PrintA();  
        a.run();  
        System.out.print("B");  
    }  
}
```

Select one:

- The program does not compile
- The output varies and is either AB or BA
- The output is always AB
- The program generates an exception at runtime
- The output is always B

NETEXAMS



NetExams

Private Edition of Internet Refresher

HOME

ABOUT
US

CONTACT

INTERVIEW

- * Name of the interviewee
- * Date

LEARNING

- * Description
- * Name of the interviewee

HOME

ACER

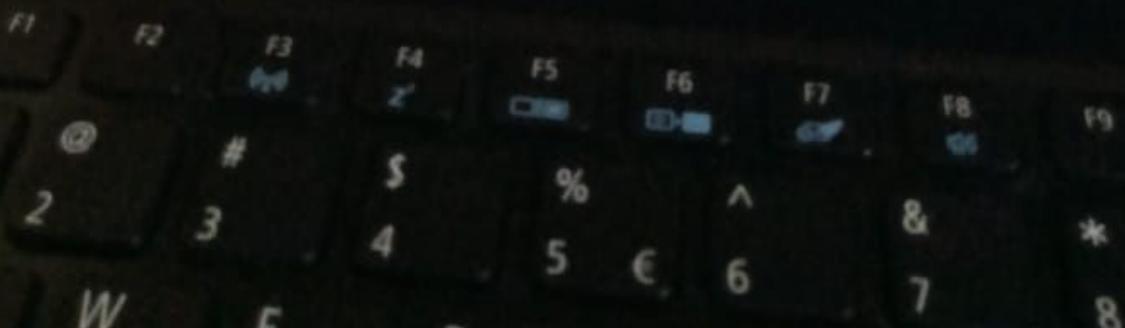
```
21  
22  
23  
24 }
```

```
System.out.println("G");
```

Select one:

- E
F
- C
F
- B
F
G
- B
F
- E
F
G

ace



```
int opt = 5;
switch (opt){
    case 1:
        System.out.println("*****");
    case 3:
        System.out.println("$$$$$");
        break;
    case 5:
        System.out.println("#####");
    default:
        System.out.println("&&&&&");
    }
}
```

Select one:

- &&&&&
- #####
- &&&&&
- #####
- Compilation error
- Run time exception

```
public class PrintA extends Thread {  
    public void run() {  
        System.out.print("A");  
    }  
}
```

which of the statement is true about the following program?

```
public class PrintB {  
    public static void main(String [] ar  
    Thread a = new PrintA();  
    a.run();  
    System.out.print("B");  
}
```

Select one:

- The output varies and is either AB or BA
- The output is always AB
- The program generates an exception at runtime
- The output is always B
- The program does not compile

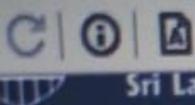
ace

What is the output of the following code:

```
public class Romeo {  
    public static void main(String args[]){  
        String s1 = "Juliet";  
        String s2 = new String(s1);  
        System.out.println(s1 == s2);  
    }  
}
```

Select one:

- a. Juliet
- b. 0
- c. true
- d. false
- e. 1



Determine the output:

```
public class Test{
    public static void main(String args[]){
        String str = null;
        if(str.length() == 0){
            System.out.print("1");
        }
        else if(str == null){
            System.out.print("2");
        }
        else{
            System.out.print("3");
        }
    }
}
```

Select one:

- a. "null" is printed
- b. "2" is printed
- c. "3" is printed
- d. "1" is printed
- e. An exception is thrown at runtime

```
4     public static void main(String[] args) {
5         int number;
6         String str;
7         try {
8             str = "5";
9             number = Integer.parseInt(str.substring(5));
10            System.out.println("A");
11        } catch (NumberFormatException e) {
12            System.out.println("B");
13        } catch (IllegalArgumentException e) {
14            System.out.println("C");
15        } catch (Exception e) {
16            System.out.println("E");
17        }
18        finally {
19            System.out.println("F");
20        }
21        System.out.println("G");
22    }
23 }
24 }
```

Select one:

- C
- F
-
- E
- F
- G
- B**
- F
- G

