

iddle X

X | i | +

 NetExam
Sri Lanka Institute of Information Technology

19 answered out of question

What is the result of the following code?

```
String city = null;
if(city.equals("Boston")) {
    System.out.print("true");
} else {
    System.out.print("false");
}finally {
    System.out.print("finally");
}
```

Select one or more:

- true
- finally, followed by the stack trace from a NullPointerException
- false
- finally
- The code does not compile.



**. finally statement that can be used to handle an exception that is not caught by any of the previous catch statements.

**. finally block can be used to handle any exception generated within a try block.

oddle

x

X

C

i

A



NetExamination

Sri Lanka Institute of Information Technology

What is the result of the following code?

```
String city = null;  
if(city.equals("Boston")) {  
    System.out.print("true");  
}else {  
    System.out.print("false");  
}finally {  
    System.out.print("finally");  
}
```

Select one or more:

- finally
- true
- false

finally, followed by the stack trace from a NullPointerException

The code does not compile.

* What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10 answer will not equal or compile error
```

Select one:

Line 4 generates a compiler error.

equal

Line 1 generates a compiler error.

An exception is thrown on line 6.

not equal

The code checks if b1.byteValue() == b2. Here's what happens:
b1.byteValue() returns the primitive byte value of b1, which is -12.
The comparison b1.byteValue() == b2 checks if this primitive byte value is equal to the Byte object b2.
However, this comparison is between a primitive byte and a Byte object, which will cause a compiler error
because the types are incompatible.



fill the blank with the correct word

When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer: **override**

Next p



20

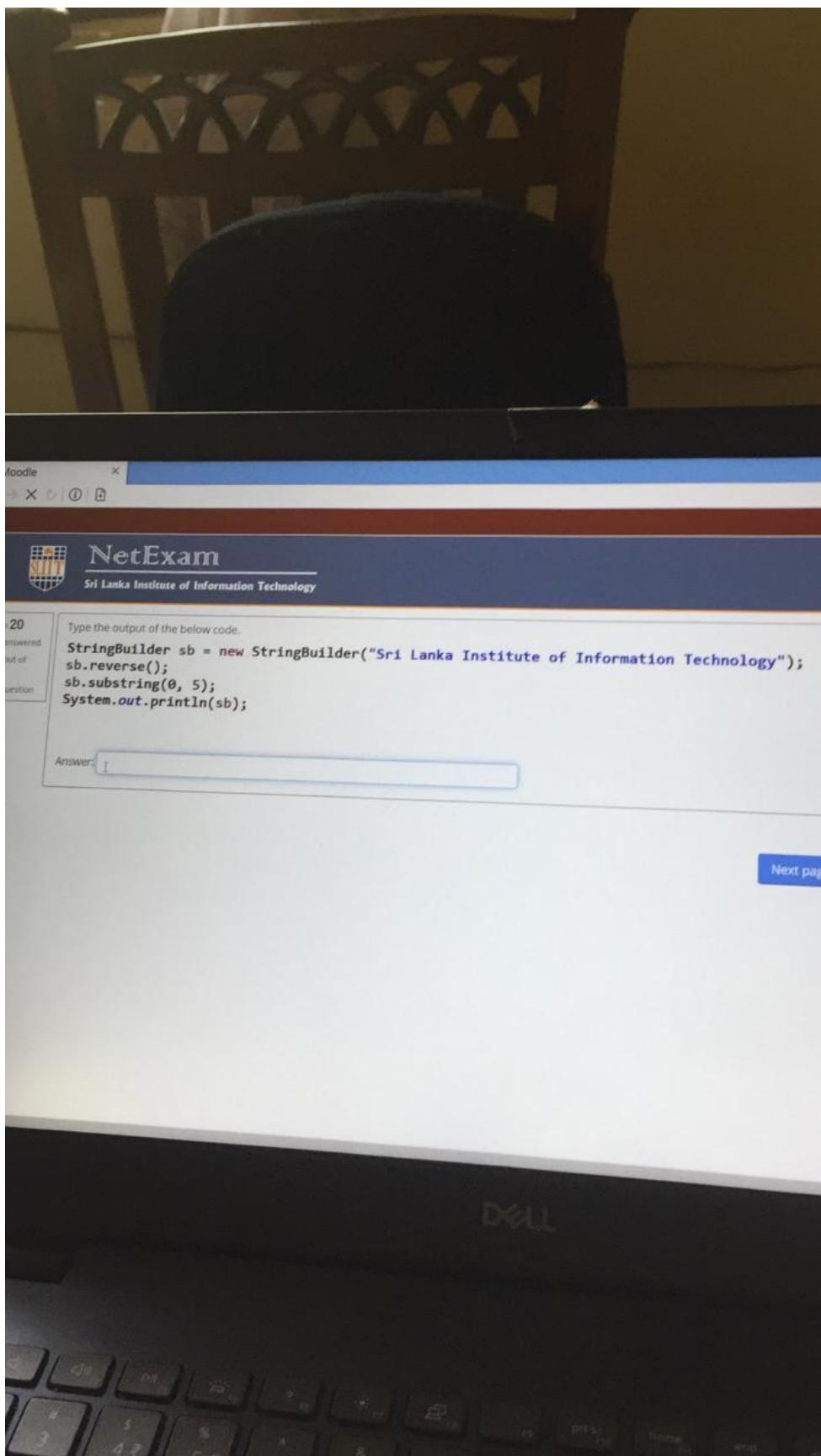
answered
out of
question

Type the output of the below code.

```
StringBuilder sb = new StringBuilder("Sri Lanka Institute of Information Technology");
sb.reverse();
sb.substring(0, 5);
System.out.println(sb);
```

Answer: **ytiritnemation noitutitsol akarL irS**

Next page



Consider the following statements to import the Scanner class in order to read a keyboard input.

- A) import java.util.Scanner;
- B) import java.util;
- C) import java.util.*;

Select one:

- All three methods are correct
- A and C are correct
- Only A is correct
- B and C are Correct
- A and B are Correct

20

answered

out of

question

Select the correct output of the following code snippet.

```
public static void main(String[] args) {  
    String result = new String("SLIIT Campus");  
    StringBuffer sb = new StringBuffer(result.toUpperCase()  
        .toLowerCase().substring(0, 6));  
    System.out.println(result);  
    System.out.println(sb);  
}
```

Select one:

- SLIIT Campus
- SLIIT Campus
- SLIIT Campus
- sliit c
- slit campus
- sliit
- SLIIT Campus
- sliit

```

static void test() {
try {
    // code creating and declaring a string variable to null
    // calling toString() method for the string variable and printing
}
finally {
    System.out.print("finally "); always executed
}
}

public static void main(String[] args) {
try {
    test(); nullPointerException is called
}
catch (Exception ex) {
    System.out.print("exception "); This should execute to handle the exception
}
}

Select one:
 a. Null finally
 b. Compilation fails
 c. Finally
 d. Finally exception
 e. Null

```

Moodle

NetExam
Sri Lanka Institute of Information Technology

Question 20
Not yet answered
Marked out of 1.00
Flag question

Type the output of the below code.

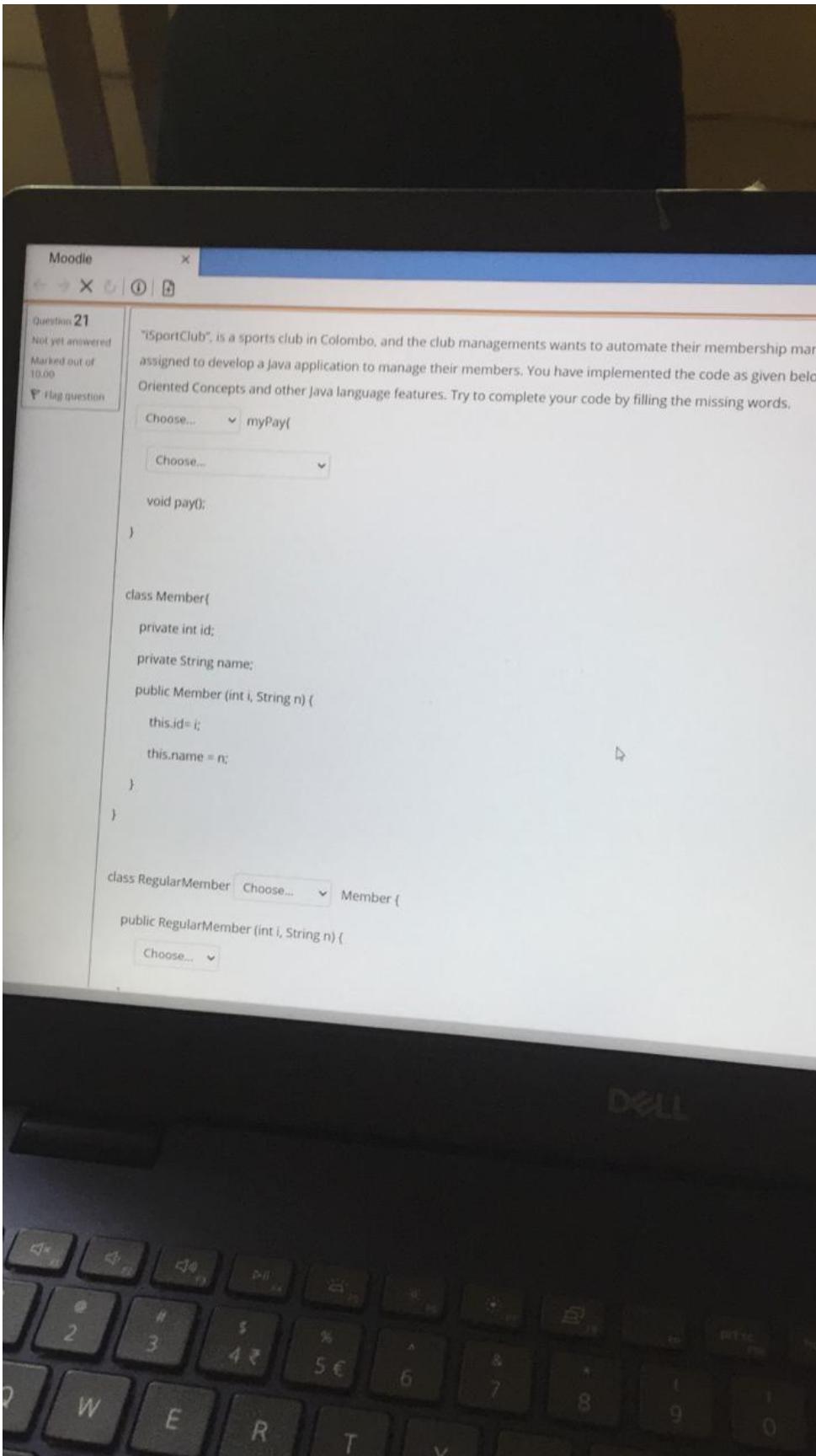
```

StringBuilder sb = new StringBuilder("Sri Lanka Institute of Information Technology");
sb.reverse();
sb.substring(0, 5);
System.out.println(sb);

```

Answer:

Next page



Type the output of the below code.

```
StringBuilder sb = new StringBuilder("Sri Lanka Institute of Information Technology");
sb.reverse();
sb.substring(0, 5); ← does not work here due to incorrect method of use
System.out.println(sb);
```

Answer: **yolohnceT noitamrofnl fo etutitsnl aknaL irS**

Next page

Sri Lanka Institute of Information Technology

Question 22
Not yet answered
Marked out of 10.00
[Flag question](#)

Complete the code by filling the missing elements.

```
public class Test2 {
    public static void main(String[] args) {
        try {
            int[] itemIds = { 34, 2, 45, 65, 3 };
            Shop s1 = new Shop(1, itemIds);
            s1.setAvailability(-2); IllegalArgumentException
            Shop s2 = null;
            try {
                System.out.println("Printing Random items");
                System.out.println(s1.getRandomItem());
                try {
                    System.out.println(s2.getRandomItem());
                } catch ( Choose... e) { nullPointerException
                    System.out.println(e.getMessage());
                }
            } catch ( Choose... e) { arrayIndexOutOfBoundsException
                System.out.println(e.getMessage());
            }
        } catch ( Choose... e) { arrayIndexOutOfBoundsException
            System.out.println(e.getMessage());
        }
    }
}
```

Finish at _____
Time left _____
MCQ QUES 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
STRUCTURE
FEEDBACK

MOODLE

NetExam
Sri Lanka Institute of Information Technology

Question 20
Not yet answered
Marked out of 1.00
Flag question

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);
```

4 1 append
1 append 4 insert
1 4 append insert

Next page

NetExam
Sri Lanka Institute of Information Technology

Question 5
Not yet answered
Marked out of 1.00
Flag question

Select the correct output for the following program.

```
class Car{
    public void printDetails(){
        System.out.println("This is a car");
    }
}
class ElectricCar{
    public void printDetails(){
        System.out.println("This is a hybrid car");
    }
}
public class Main{
    public static void main(String args[]){
        Car ECar = new Electric();
        ECar.printDetails();
    }
}
```

Select one:

- Compilation error
- System.out.println("This is a car")
- This is a hybrid car
- System.out.println("This is a hybrid car")
- This is a car

Quiz navigation

Finish attempt...
Time left: 0:39:25

MCQ QUESTIONS	1	2	3	4	5	6	7						
STRUCTURED QUESTIONS	8	9	10	11	12	13	14	15	16	17	18	19	20
FEEDBACK	21												

Next page

idle

x C | O | A

NetExam
Sri Lanka Institute of Information Technology

on 5
answered
out of
question

Select the correct output for the following program

```
class Phone{
    public void phoneDetails(){
        System.out.println("This is a phone");
    }
}
class Samsung extends Phone{
    public void phoneDetails2 (){
        System.out.println("This is a Samsung phone");
    }
}
public class Main{
    public static void main(String args[]){
        Phone myPhone = new Samsung();
        myPhone.phoneDetails();
    }
}
```

Select one:

- Compilation error
- System.out.println("This is a phone");
- This is a phone
- This is a Samsung phone
- System.out.println("This is a Samsung phone");

ASUS Vivo

1 2 3 4 5 6 7
Q W E R T Y U

Phone myPhone = new Samsung();
myPhone.phoneDetails();

&&

Phone myPhone = new Phone();
myPhone.phoneDetails();

both getting as "This is a phone."

Sri Lanka Institute of Information Technology

Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

Make all properties private and provide setter method for name and id property

Make all properties private

Make all properties private and remove setter method for id property

No need of changes

Make all methods protected

Next page

Quiz navigation

Finish attempt ...

Time left 0:55:11

MCQ QUESTIONS

STRUCTURED QUESTIONS

FEEDBACK

NetExam

Sri Lanka Institute of Information Technology

Drag and drop the most suitable word.

In Java, **classes** support multiple inheritance, but Java **Interface** do not support multiple inheritance, and when you declare any method in the Interface is by default those methods are **static** and you do not need to specify it **explicitly**.

abstract

classes

static class

Interface

abstract

final

implicitly

Next page

Quiz navigation

Finish attempt ...

Time left 0:55:11

MCQ QUESTIONS

STRUCTURED QUESTIONS

FEEDBACK

```

12     }
13 }
14 class Bottle extends Item{
15     boolean lid;
16
17     public Bottle(double capacity, double height, boolean lid) {
18         this(capacity, height);
19         System.out.println("Bottle constructor 1");
20         this.lid = lid;
21     }
22     public Bottle(double capacity, double height) {
23         System.out.println("Bottle constructor 2");
24     }
25 }
26
27 class Can extends Bottle{
28     String type;
29
30     public Can(double capacity, double height, boolean lid, String type) {
31         super(capacity, height, lid);
32         System.out.println("Can constructor");
33         this.type = type;
34     }
35 }
36
37 }
38

```

Can ob=new Can(25.0, 30, false, "Double");

What will be the output if we create the object as below?

- Select one:
- Bottle constructor 1
 - Bottle constructor 2
 - Can constructor
 - Bottle constructor 2
 - Bottle constructor 1
 - Can constructor
 - Item constructor
 - Bottle constructor 2
 - Bottle constructor 1
 - Can constructor
 - Item constructor
 - Bottle constructor 1

NetExam

Sri Lanka Institute of Information Technology

Drag and drop the most suitable word.

In java, **interface** support multiple inheritance, but java **classes** do not support multiple inheritance, and when you declare any method in the interface is by default those methods are **abstract** and you do not need to specify it **explicitly**.

Interface static class classes

Interface classes

abstract final static

explicitly implicitly

Next page

Consider the below program and select the true statement

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading..");  
    }  
}  
class programmer extends Writer {  
    programmer () {  
        super();  
    }  
    void write() {  
        System.out.print("writing..");  
    }  
}
```

Select one:

It wont compile, because class programmer must override method read()

It wont compile, because method read() must be abstract since class Writer is abstract

It wont compile, because programmer constructor is calling the super() while writer has no constructor

It will run with no errors

It will compile with no errors

Moodle

question 2

Not yet answered

Marked out of 1.00

Flag question

Select the correct output of the following program.

```
1 class Book{  
2     String name;  
3     String isbn;  
4  
5     public Book(String name, String isbn) {  
6         this.name = name;  
7         this.isbn = isbn;  
8     }  
9     @Override  
10    public String toString() {  
11        return "Book Name = " + this.name + ", Book ISBN = " + this.isbn;  
12    }  
13 }  
14 public class Ex3 {  
15     static Book book3;  
16     static Book swapParam(Book book1, Book book2){  
17         book3 = book1;  
18         book1 = book2;  
19         book2 = book3;  
20         System.out.println(book1);  
21         System.out.println(book2);  
22         return book3;  
23     }  
24     public static void main(String[] args) {  
25         Book book1 = new Book("The Other Lady Vanishes", "110-23456");  
26         Book book2 = new Book("Harry Potter", "220-23456");  
27         book2 = swapParam(book1, book2);  
28         System.out.println(book1);  
29         System.out.println(book2);  
30     }  
31 }
```

4 output lines

Select one:

Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Runtime Error



Moodle

```
16 static Book swapParam(Book book1, Book book2){  
17     book3 = book1;  
18     book1 = book2;  
19     book2 = book3;  
20     System.out.println(book1);  
21     System.out.println(book2);  
22     return book3;  
23 }  
24 public static void main(String[] args) {  
25     Book book1 = new Book("The Other Lady Vanishes", "110-23456");  
26     Book book2 = new Book("Harry Potter", "220-23456");  
27     book2 = swapParam(book1, book2);  
28     System.out.println(book1);  
29     System.out.println(book2);  
30 }  
31 }
```

Select one:

Book Name = Harry Potter, Book ISBN = 220-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
Book Name = Harry Potter, Book ISBN = 220-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456

Runtime Error

Book Name = Harry Potter, Book ISBN = 220-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456

Book Name = Harry Potter, Book ISBN = 220-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
Book Name = Harry Potter, Book ISBN = 220-23456

Book Name = Harry Potter, Book ISBN = 220-23456
Book Name = The Other Lady Vanishes, Book ISBN = 110-23456



Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

- Make all properties private
- Make all properties private and provide setter method for name and id property
- Make all properties private and remove setter method for id property
- No need of changes
- Make all methods protected

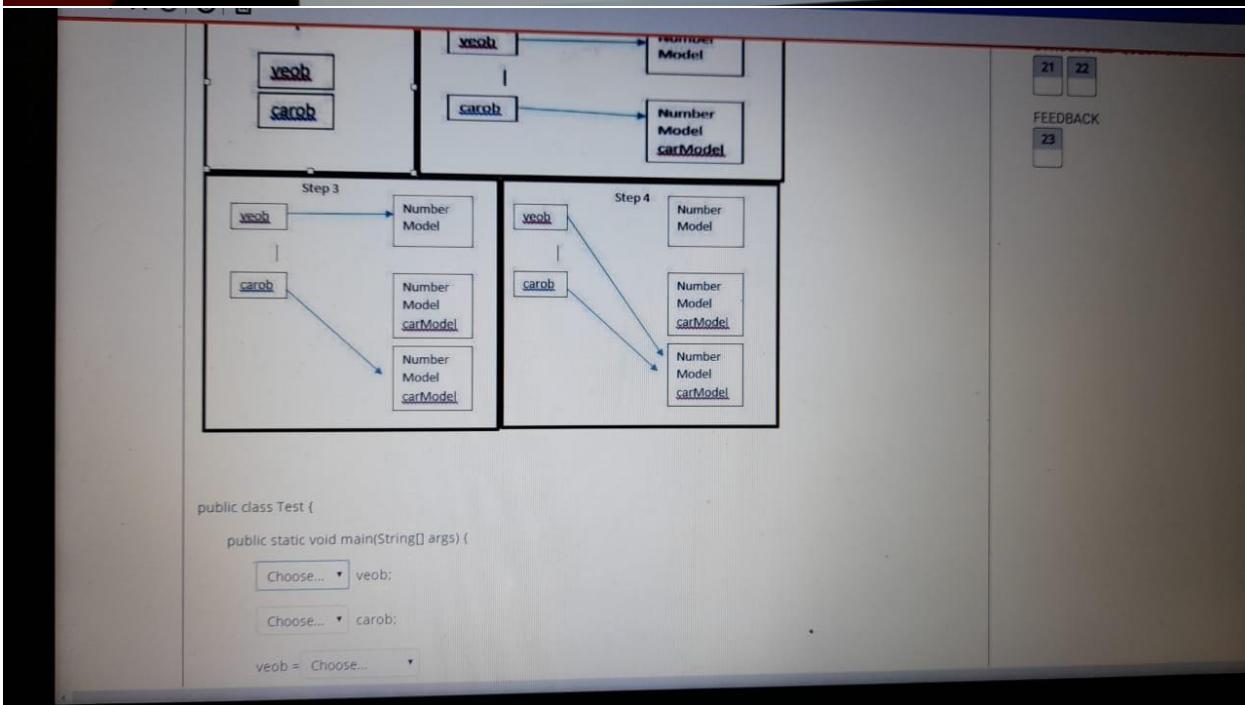
Moodle

Not yet answered
Marked out of 1.00
Flag question

```
abstract class Writer {
    abstract void write();
    void read() {
        System.out.print("reading..");
    }
}
class programmer extends Writer {
    programmer () {
        super();
    }
    void write() {
        System.out.print("writing..");
    }
}
```

Select one:

- It wont compile, because class programmer must override method read ()
- It wont compile, because programmer constructor is calling the super() while Writer has no constructor defined
- It will compile with no errors
- It wont compile, because method read() must be abstract since class Writer is abstract
- It will run with no errors



Moodle

Question 3
Not yet answered
Marked out of 1.00
Flag question

Assume that you are given two classes as below.

```
public class Vehicle {  
    private int Number;  
    private String model;  
}  
public class Car extends Vehicle{  
    private String carModel;  
}  
public class Van extends Vehicle{  
    private String vanModel;  
}
```

Fill the code with most suitable statement according to the memory map given. Each diagram represent set of steps after the statements.

The diagram illustrates the state of memory across four steps:

- Step 1:** Shows two objects: `veob` and `carob`. `veob` contains a `Number` field and a `model` field. `carob` contains a `Number` field and a `model` field.
- Step 2:** The `model` field of `veob` is assigned the value "Number Model". The `model` field of `carob` is assigned the value "Number Model carModel".
- Step 3:** The `model` field of `veob` is assigned the value "Number Model". The `model` field of `carob` is assigned the value "Number Model carModel". A new object `Number Model carModel` is created and assigned to the `model` field of `carob`.
- Step 4:** The `model` field of `veob` is assigned the value "Number Model". The `model` field of `carob` is assigned the value "Number Model carModel". Both `veob` and `carob` now point to the same external object "Number Model carModel".

```
public class Test {  
    public static void main(String[] args) {  
        // Code to be filled in  
    }  
}
```

Select the correct output for the following program.

```
class A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 2:4:1:3
- 1:3:1:3
- 1:3:2:4
- 1:4:1:4

ASUS VIV

Not yet answered

Marked out of
1.00

Flag question

```
1 class A{
2     public A() {
3         System.out.println("A()");
4     }
5 }
6 class B extends A{
7     public B() {
8         System.out.println("B()");
9     }
10    public B(int no) {
11        System.out.println("B(int no)");
12    }
13 }
14 class C extends B{
15     public C() {
16         super(20);
17         System.out.println("C()");
18     }
19    public C(int no) {
20        this();
21        System.out.println("C(int no)");
22    }
23 }
24 class D extends C{
25     public D() {
26         super(10);
27         System.out.println("D()");
28    }
29 }
30 public class Ex1 {
31     public static void main(String[] args) {
32         new D();
33     }
34 }
```

A()
B(int no)
C()
C(int no)
D()

Select one:

- A()
- B()
B(int no)
- C()
- C(int no)
- D()

- Compile Error



NetExam

Sri Lanka Institute of Information Technology

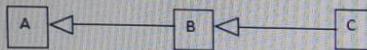
Question 5

Not yet answered

Marked out of
1.00

Flag question

Classes A, B and C are having the inheritance relationship as given below.



If public void sumof2() is a method signature of a method in class B, select the correct way to implement method **overriding**.

Select one:

- Implement public void sumof2() in class C
- Implement public void sumof2() in class A
- Implement public void sumof22() in class C
- Implement public void sumof2() in class B
- Implement public int sumof2() in class C

Next page

Consider the below program and select the true statement

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading..");  
    }  
}  
class programmer extends Writer {  
    programmer () {  
        super();  
    }  
    void write() {  
        System.out.print("writing..");  
    }  
}
```

Select one:

It wont compile, because class programmer must override method read()

It wont compile, because method read() must be abstract since class Writer is abstract

It wont compile, because programmer constructor is calling the super() while writer has no constructor

It will run with no errors

It will compile with no errors

Moodle

question 2

Not yet answered

Marked out of 1.00

Flag question

Select the correct output of the following program.

```
1 class Book{  
2     String name;  
3     String isbn;  
4  
5     public Book(String name, String isbn) {  
6         this.name = name;  
7         this.isbn = isbn;  
8     }  
9     @Override  
10    public String toString() {  
11        return "Book Name = " + this.name + ", Book ISBN = " + this.isbn;  
12    }  
13 }  
14 public class Ex3 {  
15     static Book book3;  
16     static Book swapParam(Book book1, Book book2){  
17         book3 = book1;  
18         book1 = book2;  
19         book2 = book3;  
20         System.out.println(book1);  
21         System.out.println(book2);  
22         return book3;  
23     }  
24     public static void main(String[] args) {  
25         Book book1 = new Book("The Other Lady Vanishes", "110-23456");  
26         Book book2 = new Book("Harry Potter", "220-23456");  
27         book2 = swapParam(book1, book2);  
28         System.out.println(book1);  
29         System.out.println(book2);  
30     }  
31 }
```

harry poter
other lady
other lady
other lady

Select one:

Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Runtime Error



Moodle

```
16 static Book swapParam(Book book1, Book book2){  
17     book3 = book1;  
18     book1 = book2;  
19     book2 = book3;  
20     System.out.println(book1);  
21     System.out.println(book2);  
22     return book3;  
23 }  
24 public static void main(String[] args) {  
25     Book book1 = new Book("The Other Lady Vanishes", "110-23456");  
26     Book book2 = new Book("Harry Potter", "220-23456");  
27     book2 = swapParam(book1, book2);  
28     System.out.println(book1);  
29     System.out.println(book2);  
30 }  
31 }
```

Select one:

Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Runtime Error
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456
 Book Name = Harry Potter, Book ISBN = 220-23456
 Book Name = The Other Lady Vanishes, Book ISBN = 110-23456



Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

- Make all properties private
- Make all properties private and provide setter method for name and id property
- Make all properties private and remove setter method for id property
- No need of changes
- Make all methods protected

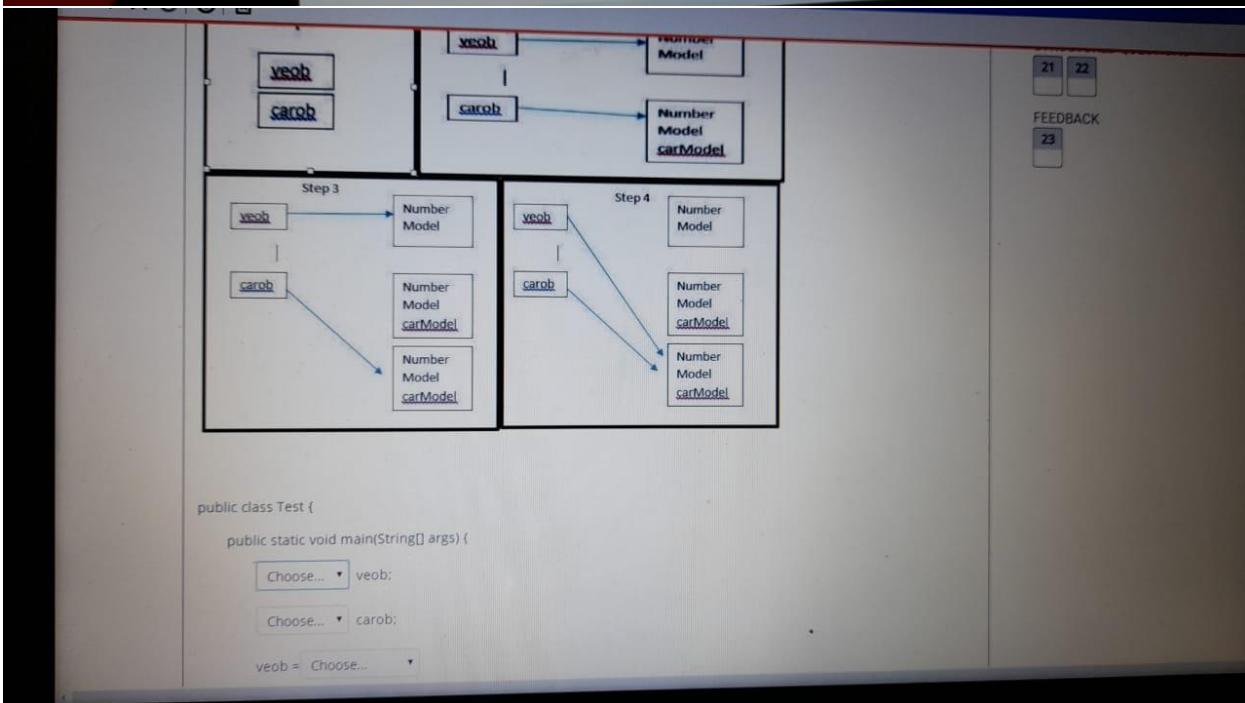
Moodle

Not yet answered
Marked out of 1.00
Flag question

```
abstract class Writer {
    abstract void write();
    void read() {
        System.out.print("reading..");
    }
}
class programmer extends Writer {
    programmer () {
        super();
    }
    void write() {
        System.out.print("writing..");
    }
}
```

Select one:

- It wont compile, because class programmer must override method read()
- It wont compile, because programmer constructor is calling the super() while Writer has no constructor defined
- It will compile with no errors
- It wont compile, because method read() must be abstract since class Writer is abstract
- It will run with no errors



Vehicle veob;
Car carob;

veob = new Vehicle();
carob = new Car();

carob = new Car();

veob = carob;

Moodle

Question 3
Not yet answered
Marked out of 1.00
Flag question

Assume that you are given two classes as below.

```
public class Vehicle {
    private int Number;
    private String model;
}
public class Car extends Vehicle{
    private String carModel;
}
public class Van extends Vehicle{
    private String vanModel;
}
```

Fill the code with most suitable statement according to the memory map given. Each diagram represent set of steps after the statements.

The diagram illustrates the state of memory across four steps:

- Step 1:** Two objects are created: `veob` and `carob`. They are shown in separate boxes.
- Step 2:** The `veob` object has a reference to a `Number Model`. The `carob` object also has a reference to a `Number Model`, which is labeled `Number Model carModel`.
- Step 3:** The `veob` object now points to its own `Number Model`. The `carob` object points to a new `Number Model carModel` object, and there is also a separate `Number Model carModel` object.
- Step 4:** Both `veob` and `carob` now point to their respective `Number Model carModel` objects.

Below the diagram, the code is partially visible:

```
public class Test {
    public static void main(String[] args) {
        // ...
    }
}
```

Vehicle veob;
Car carob;

veob = new Vehicle();
carob = new Car();

carob = new Car();

veob = carob;

Select the correct output for the following program.

```
class A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 2:4:1:3
- 1:3:1:3
- 1:3:2:4
- 1:4:1:4

ASUS VIV

Not yet answered

Marked out of
1.00

Flag question

```
1 class A{
2     public A() {
3         System.out.println("A()");
4     }
5 }
6 class B extends A{
7     public B() {
8         System.out.println("B()");
9     }
10    public B(int no) {
11        System.out.println("B(int no)");
12    }
13 }
14 class C extends B{
15     public C() {
16         super(20);
17         System.out.println("C()");
18     }
19    public C(int no) {
20        this();
21        System.out.println("C(int no)");
22    }
23 }
24 class D extends C{
25     public D() {
26         super(10);
27         System.out.println("D()");
28    }
29 }
30 public class Ex1 {
31     public static void main(String[] args) {
32         new D();
33     }
34 }
```

A()
B(int no)
C()
C(int no)
D()

Select one:

- A()
- B()
B(int no)
- C()
- C(int no)
- D()

- Compile Error



NetExam

Sri Lanka Institute of Information Technology

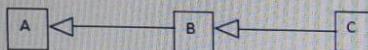
Question 5

Not yet answered

Marked out of
1.00

Flag question

Classes A, B and C are having the inheritance relationship as given below.



If public void sumof2() is a method signature of a method in class B, select the correct way to implement method **overriding**.

Select one:

- Implement public void sumof2() in class C
- Implement public void sumof2() in class A
- Implement public void sumof2() in class C
- Implement public void sumof2() in class B
- Implement public int sumof2() in class C

Next page

Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

- Make all properties private
- Make all properties private and remove setter method for id property
- Make all methods protected
- No need of changes
- Make all properties private and provide setter method for name and id property

Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

- Make all properties private
- Make all properties private and remove setter method for id property
- Make all methods protected
- No need of changes
- Make all properties private and provide setter method for name and id property

Select the correct output for the following program.

```
Class A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 2:4,1:3
- 1:3,1:3,
- 1:3,2:4
- 1:4,1:4,
- 2:3,1:4

Select the correct output of the following program?

```
1 class A{  
2     public A() {  
3         System.out.println("A()");  
4     }  
5 }  
6 class B extends A{  
7     public B() {  
8         System.out.println("B()");  
9     }  
10    public B(int no) {  
11        System.out.println("B(int no)");  
12    }  
13 }  
14 class C extends B{  
15    public C() {  
16        super(20);  
17        System.out.println("C()");  
18    }  
19    public C(int no) {  
20        this();  
21        System.out.println("C(int no)");  
22    }  
23 }  
24 class D extends C{  
25    public D() {  
26        super(10);  
27        System.out.println("D()");  
28    }  
29 }  
30 public class Ex1 {  
31     public static void main(String[] args) {  
32         new D();  
33     }  
34 }
```

Select one:

- A()
B(int no)
C()

- A()
- B(int no)
- C()
- C(int no)
- D()

```
21     System.out.println("C(int no)");
22 }
23 }
24 class D extends C{
25     public D() {
26         super(10);
27         System.out.println("D()");
28     }
29 }
30 public class Ex1 {
31     public static void main(String[] args) {
32         new D();
33     }
34 }
```

Select one:

- A()
B(int no)
- C()
- C(int no)
- D()
- Compile Error
- A()
B()
 - B(int no)
 - C()
 - C(int no)
 - D()
- Constructor Cyclic dependency runtime error
- A()
B(int no)
B()
C(int no)
C()
D()

NetExam
Sri Lanka Institute of Information Technology

4 answered out of 4 question

Drag and drop the most suitable word.

In java, **interface** support multiple inheritance, but java **classes** do not support multiple inheritance, and when you declare any method in the interface is by default those methods are **abstract** and you do not need to specify it **explicity**.

Interface classes static class

Interface classes

abstract final static

implicitly explicitly

Next page

Sri Lanka Institute of Information Technology

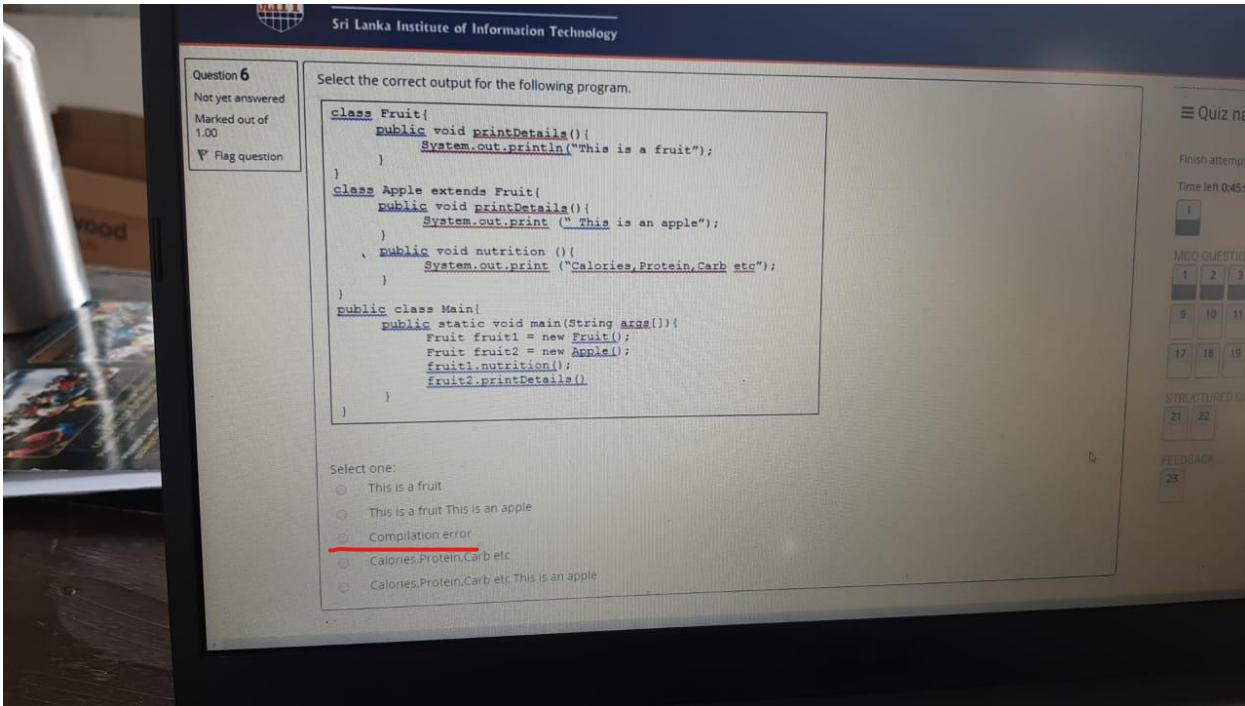
Question 6 Not yet answered Marked out of 1.00 Flag question

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}
class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}
public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails();
    }
}
```

Select one:

- This is a fruit
- This is a fruit This is an apple
- Compilation error
- Calories,Protein,Carb etc
- Calories,Protein,Carb etc This is an apple



Sri Lanka Institute of Information Technology

Question 6
Not yet answered
Marked out of 1.00
 Flag question

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}

class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}

public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails();
    }
}
```

Select one:

- This is a fruit
- This is a fruit This is an apple
- Compilation error
- Calories,Protein,Carb etc
- Calories,Protein,Carb etc This is an apple

Quiz na
Finish attempt
Time left 0:45:00
MCQ QUESTIO
1 2 3
9 10 11
17 18 19
STRUCTURED Q
21 22
FEEDBACK
23

Consider the below program and select the true statement

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading..");  
    }  
}  
class programmer extends Writer {  
    programmer () {  
        super();  
    }  
    void write() {  
        System.out.print("writing..");  
    }  
}
```

Select one:

- It wont compile, because programmer constructor is calling the super() while Writer is abstract
- It wont compile, because class programmer must override method read()
- It will run with no errors
- It wont compile, because method read() must be abstract since class Writer is abstract
- It will compile with no errors

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

```
    Vehicle veob;
```

```
    Vehicle carob;
```

```
    veob = new Vehicle();
```

```
    carob = new Car();
```

```
    carob = new Car();
```

```
    veob = carob;
```

```
}
```

NetExam

Sri Lanka Institute of Information Technology

Examine the below code.

```
public class Student {  
    public int id;  
    protected String name;  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

What changes need to be done so that the student class comply the Encapsulation?

Select one or more:

Make all properties private
 No need of changes
 Make all methods protected
 Make all properties private and remove setter method for id property
 Make all properties private and provide setter method for name and id property

 NetExam
Sri Lanka Institute of Information Technology

Select the correct method signatures to implement cloning objects of following

```
class Patient{  
    int age;  
    String name;  
    public Patient(){  
        this.age = 0;  
        this.name = "No name";  
    }  
    public Patient (int age, String name) {  
        this.age = age;  
        this.name = name;  
    }  
}
```

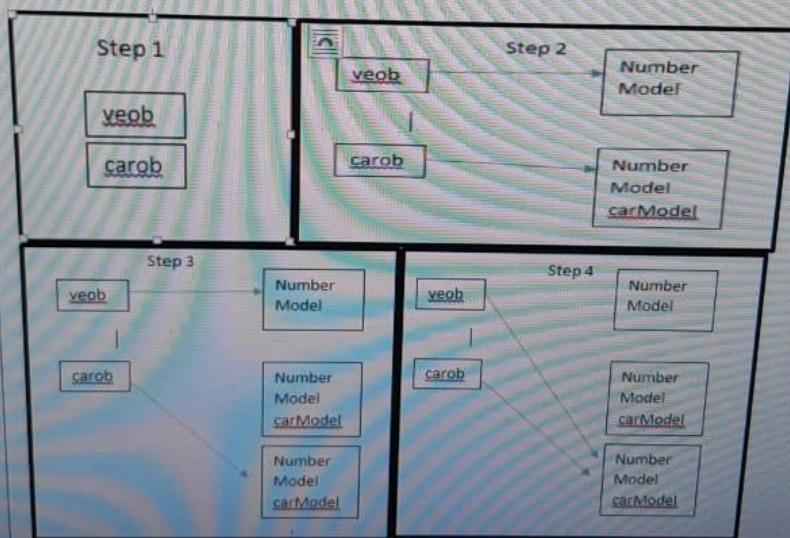
Select one:

- Implement public Patient(String p)
- Implement public Patient(Patient p)
- Implement public void clonePatient(int age, String name)
- Implement public void Patient(Patient p)
- Implement public void clone(Patient p)

Assume that you are given two classes as below.

```
public class Vehicle {  
    private int Number;  
    private String model;  
}  
public class Car extends Vehicle{  
    private String carModel;  
}  
public class Van extends Vehicle{  
    private String vanModel;  
}
```

Fill the code with most suitable statement according to the memory map given. Each diagram represents steps after the statements.



Vehicle veob;
Car carob;

veob = new Vehicle();
carob = new Car();

carob = new Car();

veob = carob;

Moodle

Sri Lanka Institute of Information Technology

Question 5
Not yet answered
Marked out of 1.00
 Flag question

Select the correct output for the following program.

```
class A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 1:3,1:3,
- 2:4,1:3
- 1:4,1:4,
- 2:3,1:4,

DELL

*

Powered
by

Quiz navigation

Finish attempt ...

Time left 0:46:48

MQ QUESTIONS

1	2	3
8	9	10
15	16	17

STRUCTURED QUIZZES

21	22
----	----

FEEDBACK

23

```
class Parent {  
    Parent (int x) {  
        System.out.println("Super");  
    }  
}  
public class Child extends Parent {  
    Child () {  
        // Line X  
        System.out.println("Sub 2");  
    }  
}
```

Which statement, when inserted at Line X, enables the code to compile?

Select one:

Parent(10);
 super, Parent (10);
 this.Parent(10);
 this(10);
 super(10);

Powered
by

Quiz navigation

Finish attempt ...

Time left 0:46:48

MQ QUESTIONS

1	2	3
8	9	10
15	16	17

STRUCTURED QUIZZES

21	22
----	----

FEEDBACK

23

```
class Parent {  
    Parent (int x) {  
        System.out.println("Super");  
    }  
}  
public class Child extends Parent {  
    Child () {  
        // Line X  
        System.out.println("Sub 2");  
    }  
}
```

Which statement, when inserted at Line X, enables the code to compile?

Select one:

Parent(10);
 super, Parent (10);
 this.Parent(10);
 this(10);
 super(10);



Answered
of
question

Drag and drop the most suitable word.

In java, **Interface** support multiple inheritance, but java **classes** do not support multiple inheritance, and when you declare any method in the interface is by default those methods are **abstract** and you do not need to specify it **explicitly**.

classes **Interface** **static class**

Interface **classes**

static **final** **abstract**

explicitly **implicitly**

[Next page](#)

Question 4

Not yet answered
Marked out of
1.00

[Flag question](#)

Examine the below classes.

```
1 public class Item {  
2     double capacity;  
3     double height;  
4     public Item() {  
5     }  
6     public Item(double capacity, double height) {  
7         super();  
8         System.out.println("Item constructor");  
9         this.capacity = capacity;  
10        this.height = height;  
11    }  
12 }  
13 class Bottle extends Item{  
14     boolean lid;  
15     public Bottle(double capacity, double height, boolean lid) {  
16         super(capacity, height);  
17         System.out.println("Bottle constructor 1");  
18         this.lid = lid;  
19     }  
20     public Bottle(double capacity, double height) {  
21         super(capacity, height);  
22         System.out.println("Bottle constructor 2");  
23     }  
24 }  
25 class Can extends Bottle{  
26     String type;  
27     public Can(double capacity, double height, boolean lid, String type) {  
28         super(capacity, height, lid);  
29         System.out.println("Can constructor");  
30         this.type = type;  
31     }  
32 }  
33 Can ob=new Can(25.0, 30, false, "Double");
```

What will be the output if we create the object as below?

`Can ob=new Can(25.0, 30, false, "Double");`

```

25     System.out.println("Bottle constructor 2");
26 }
27
28 }
29 class Can extends Bottle{
30     String type;
31
32 public Can(double capacity, double height, boolean lid, String type) {
33     super(capacity, height, lid);
34     System.out.println("Can constructor");
35     this.type = type;
36 }
37
38 }

```

What will be the output if we create the object as below?

Can ob=new Can(25.0, 30, false, "Double");

Select one:

- Item constructor
Bottle constructor 2
Can constructor
- Bottle constructor 1
Bottle constructor 2
Can constructor
- Item constructor
Bottle constructor 2
Bottle constructor 1
Can constructor
- Bottle constructor 2
Bottle constructor 1
Can constructor
- Item constructor
Bottle constructor 1
Can constructor

What is true about constructor?

- A The constructor is used to initialize the object when it is declared
- B. The constructor has a return value, and return type
- C The constructors can have the same name as the class
- D There can be default constructor or constructors with parameters
- E. Inside every subclass, super class constructor should be call by the developer

not always only if superclass has overload constructor

Select one or more:

- D
- A
- B
- E
- C

NetExams

Sri Lanka Institute of Information Technology

Drag and drop the most suitable word.

When override the methods the return types of both methods should be same or the overridden method return type should be **a sub class** of the return type of superclass method. And methods signature should be same. Access modifiers of the method should be **same** **same or wider**

Object class **a sub class** **a super class**

same **lower** **same or wider**

Next page

 |  | 

NetExam

Sri Lanka Institute of Information Technology

Select the correct statement to assign the value of number1 to number2 given below.

```
double number1=12.12;  
float number2;
```

Select one:

- (double) number2 = number1;
- number1 = (float) number2;
- number1 = (double) number2;
- number2 = (float) number1;
- number2 = number1;



Sri Lanka Institute of Information Technology

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}
class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}
public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails()
    }
}
```

Select one:

- This is a fruit This is an apple
- Calories,Protein,Carb etc
- Compilation error
- This is a fruit
- Calories,Protein,Carb etc This is an apple

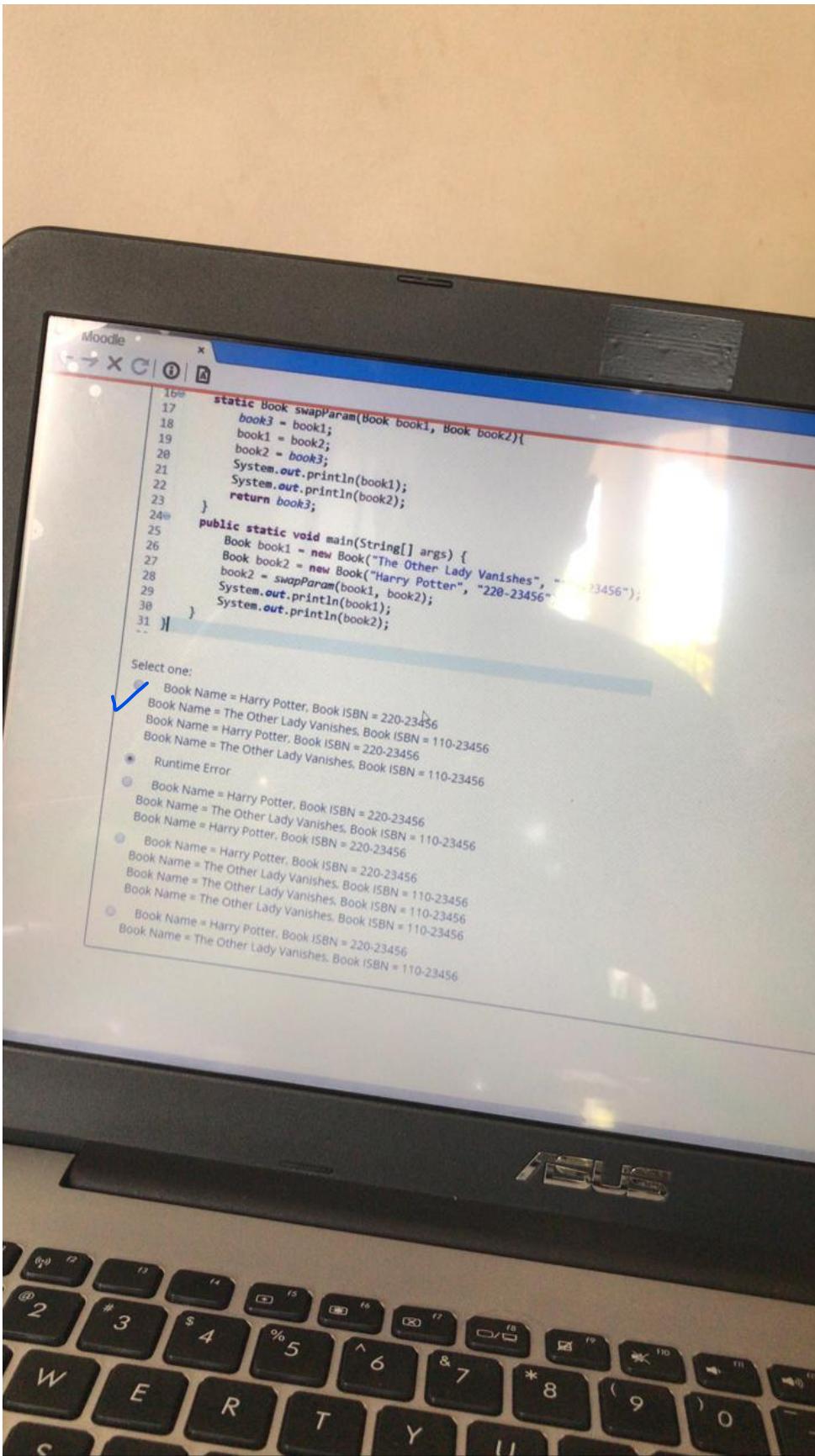
Select the correct statement to assign the value of number1 to number2 given below.

```
double number1=12.12;
```

```
float number2;
```

Select one:

- number1 = (float) number2;
- number2 = (float) number1;
- number2 = number1;
- (double) number2 = number1;
- number1 = (double) number2;



 NetExam
Sri Lanka Institute of Information Technology

Select the correct statement to assign the value of number1 to number2 given below.

```
double number1=12.12;  
float number2;
```

Select one:

- (double) number2 = number1;
- number1 = (float) number2;
- number1 = (double) number2;
- number2 = (float) number1;
- number2 = number1;

Multiple inheritance means,

Select one:

- a.
Two answers listed are correct
- b.
one class inheriting from more super classes
- c.
None of the mention
- d.
more classes inheriting from more super classes
- e.
more classes inheriting from one super class



Sri Lanka Institute of Information Technology

Question 4

Not yet answered
Marked out of
1.00

Flag question

What is true about constructor?

- A. the constructor is used to initialize the object when it is declared
- B. The constructor has a return value, and return type
- C. The constructors can have the same name as the class
- D. There can be default constructor or constructors with parameters
- E. Inside every subclass, super class constructor should be call by the developer

Select one or more:

- A
- E
- C
- B
- D

Next page

**Question 4**

Not yet answered

Marked out of
1.00

Flag question

What is true about constructor?

- A. The constructor is used to initialize the object when it is declared
- B. The constructor has a return value, and return type
- C. The constructors can have the same name as the class
- D. There can be default constructor or constructors with parameters
- E. Inside every subclass, super class constructor should be call by the developer

not always only if superclass has
overload constructor

Select one or more:

- A
- E
- C
- B
- D

Next page



NetExamination

Sri Lanka Institute of Information Technology

Question 5

Not yet answered
Marked out of
1.00

Flag question

Select the correct method signatures to implement

```
class Patient{  
    int age;  
    String name;  
    public Patient(){  
        this.age = 0;  
        this.name = "No name";  
    }  
    public Patient (int age, String name) {  
        this.age = age;  
        this.name = name;  
    }  
}
```

Select one:

- Implement public void clone(Patient p)
- Implement public Patient(String p)
- Implement public void clonePatient(int age, String name)
- Implement public void Patient(Patient p)
- Implement public Patient(Patient p)

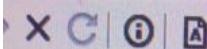
Select the correct output for the following program.

```
Class Car{  
    String type;  
    double mileage;  
    Public Car(String type, double mil){  
        this.mileage = mil;  
        this.type = type;  
    }  
  
    public void updateMileage(Car c){  
        mileage = c.mileage;  
    }  
    public void printMileage(){  
        System.out.print(this.mileage + ", ");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        Car oldCar = new Car("Toyota", 23456);  
        Car newCar = new Car("Honda", 1200);  
  
        oldCar.updateMileage(newCar);  
        newCar.updateMileage(newCar);  
  
        oldCar.printMileage();  
        newCar.printMileage();  
    }  
}
```

Select one:

- 1200, 23456,
- 23456, 1200,
- 1200, 0,
- 1200, 1200,
- 23456, 23456,

Code



Sri Lanka Institute of Information Technology

7

answered
out of
question

What is the **Polymorphism** technique used in below code?

```
class Father{
    public void printDetails(){
        System.out.println("I am Mr. Perera");
    }
}
class Son extends Father{
    public void printDetails(){
        System.out.println("I am Master Perera");
    }
}
public class Main{
    public static void main(String args[]){
        Son tharindu = new Son();
        tharindu.printDetails();
    }
}
```

Select one:

- Overloading
- Inheritance
- Overriding
- Method Chaining
- Abstraction



Moodle

Not yet answered
Marked out of 1.00
Flag question

```
class A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 1:3:2:4
- 2:4:1:3
- 1:3:1:3
- 2:3:1:4
- 1:4:1:4

Not yet answered
Marked out of
1.00
 Flag question

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}
class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}
public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails();
    }
}
```

Select one:

- This is a fruit
- Compilation error
- This is a fruit This is an apple
- Calories,Protein,Carb etc
- Calories,Protein,Carb etc This is an apple



on 6
answered
out of
g question

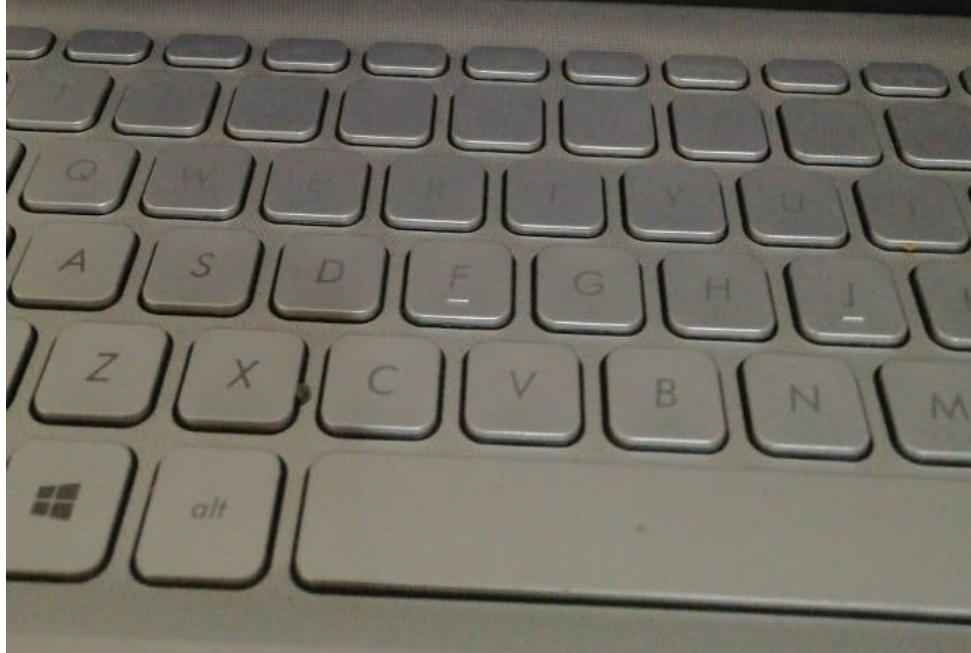
Select the correct output for the following program

```
class Phone{
    public void phoneDetails(){
        System.out.println("This is a phone");
    }
}
class Samsung extends Phone{
    public void phoneDetails2 (){
        System.out.println("This is a Samsung phone");
    }
}
public class Main{
    public static void main(String args[]){
        Phone myPhone = new Samsung();
        myPhone.phoneDetails();
    }
}
```

Select one:

- This is a Samsung phone
- System.out.println("This is a Samsung phone");
- This is a phone
- Compilation error
- System.out.println("This is a phone");

ASUS VivoBook





on 6
It answered
d out of
g question

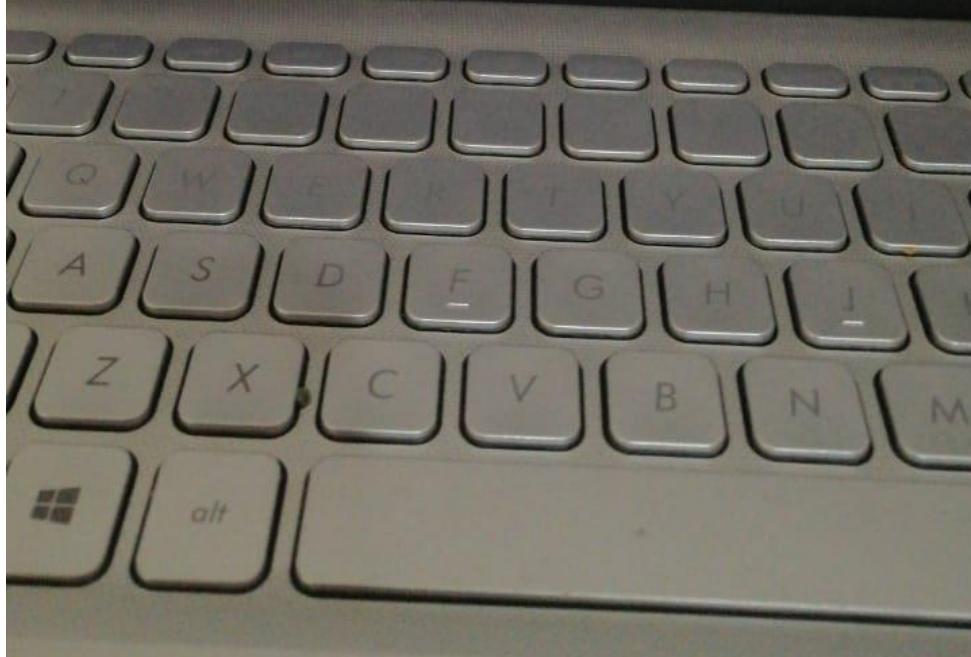
Select the correct output for the following program

```
class Phone{
    public void phoneDetails(){
        System.out.println("This is a phone");
    }
}
class Samsung extends Phone{
    public void phoneDetails2 (){
        System.out.println("This is a Samsung phone");
    }
}
public class Main{
    public static void main(String args[]){
        Phone myPhone = new Samsung();
        myPhone.phoneDetails();
    }
}
```

Select one:

- This is a Samsung phone
- System.out.println("This is a Samsung phone");
- This is a phone
- Compilation error
- System.out.println("This is a phone");

ASUS VivoBook



Not yet answered
Marked out of 1.00
Flag question

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading..");  
    }  
}  
class programmer extends Writer {  
    programmer () {  
        super();  
    }  
    void write() {  
        System.out.print("writing..");  
    }  
}
```

Select one:

It wont compile, because class programmer must override method read ()

It will compile with no errors

It will run with no errors

It wont compile, because method read() must be abstract since class Writer is abstract.

It wont compile, because programmer constructor is calling the super() while Writer has no constructor defined

NetExam
Sri Lanka Institute of Information Technology

Section 3
Not yet answered
Marked out of 1.00
Flag question

Drag and drop the most suitable word.

In java, **Interface** support multiple inheritance, but java **classes** do not support multiple inheritance, and when you declare any method in the interface is by default those methods are **static** and you do not need to specify it **explicitly**.

static class classes

Interface
abstract final
implicitly

Next page

Finish attempt ...
Time left 0:46:55

MCQ QUESTIONS
1 2 3 4
9 10 11 12
17 18 19 20

STRUCTURED QUESTIONS
21 22

FEEDBACK
23



NetExam

Sri Lanka Institute of Information Technology

Powered
by SLIIT
Question

Select when the manual casting is required for a program with the following:

```
String number1= 'abc';  
float number2 = 12.54f;  
double number3 = 25;  
int number4;
```

Select one:

- When assigning the value of number2 to number3
- When assigning the value of number4 to number2
- When assigning the value of number1 to number3
- When assigning the value of number2 to number4
- When assigning the value of number1 to number2



Section 4
Not yet attempted
Marked out of
100
100%
Reattempt

Consider the below program and select the true statement

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading...");  
    }  
}  
class programmer extends Writer {  
    programmer() {  
        super();  
    }  
    void write() {  
        System.out.print("writing...");  
    }  
}
```

- Ques 1/20
- If an instance of the Writer class is created, it will print "reading..." when its read() method is called.
 - If an instance of the programmer class is created, it will print "writing..." when its write() method is called.
 - If an instance of the Writer class is created, it will print "writing..." when its write() method is called.
 - If an instance of the programmer class is created, it will print "reading..." when its read() method is called.

Ques 1/20

Final answer

Was it helpful?

Helpful

Not helpful

Very not helpful

Worst ever!

Report

```
        double mileage;
    public Car(String type, double mil){
        this.mileage = mil;
        this.type = type;
    }

    public void updateMileage(Car c){
        mileage = c.mileage;
    }
    public void printMileage(){
        System.out.print(this.mileage + ", ");
    }

}

public class Main{
    public static void main(String args[]){
        Car oldCar = new Car("Toyota",23456);
        Car newCar = new Car("Honda",1200);

        oldCar.updateMileage(newCar);
        newCar.updateMileage(newCar);

        oldCar.printMileage();
        newCar.printMileage();
    }

}
```

object created by the car class as a variable

Select one:

- 23456, 1200,
- 1200, 0,
- 23456, 23456,
- 1200, 1200,
- 1200, 23456,

Select the correct output for the following program.

```
class Car{  
    String type;  
    double mileage;  
    public Car(String type, double mil){  
        this.mileage = mil;  
        this.type = type;  
    }  
  
    public void updateMileage(Car c){  
        mileage = c.mileage;  
    }  
    public void printMileage(){  
        System.out.print(this.mileage + ", ");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        Car oldCar = new Car("Toyota", 23456);  
        Car newCar = new Car("Honda", 1200);  
  
        oldCar.updateMileage(newCar);  
        newCar.updateMileage(newCar);  
  
        oldCar.printMileage();  
        newCar.printMileage();  
    }  
}
```

Select one:

- 23456, 1200,
- 1200, 0,
- 23456, 23456,
- 1200, 1200. ✓

File X C | ① | A it20

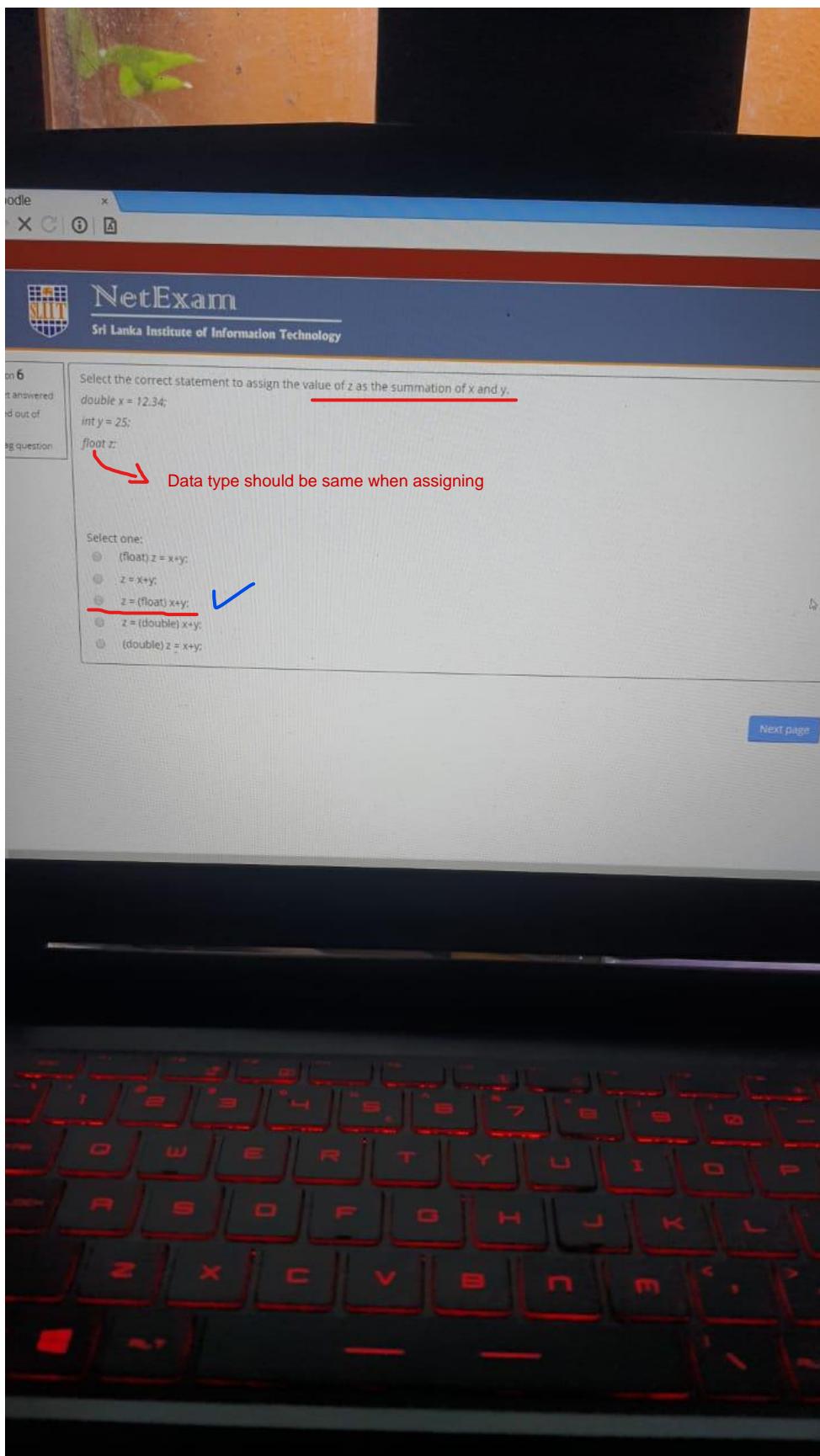
NetExam
Sri Lanka Institute of Information Technology

When an overridden method is called from within a subclass, it will always refer to the version of that method defined by the

Select one:

- a. Super class
- b. None of the mention
- c. Interpreter will choose randomly
- d. Subclass
- e. Compiler will choose randomly

[Next page](#)



Consider the bellow program and select the true statement

```
abstract class Writer {  
    abstract void write();  
    void read() {  
        System.out.print("reading..");  
    }  
}  
class programmer extends Writer {  
    programmer () {  
        super();  
    }  
    void write() {  
        System.out.print("writing..");  
    }  
}
```

Select one:

- It will compile with no errors
- It wont compile, because programmer constructor is calling the super() while Writer has no constructor defined
- It wont compile, because class programmer must override method read()
- It wont compile, because method read() must be abstract since class Writer is abstract
- It will run with no errors

Next pa

Question 5
Not yet answered
Marked out of
1.00
 Flag question

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}
class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}
public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails();
    }
}
```

Select one:

- This is a fruit This is an apple
- Calories,Protein,Carb etc This is an apple
- Compilation error
- This is a fruit
- Calories,Protein,Carb etc

≡ Q

Finish

Time

I

MCQ

1

8

15

STRU

21

FEED

23

Question 8
Not yet answered
Marked out of
1.00
 Flag question

Sri Lanka Institute of Information Technology

Select the correct output for the following program.

```
class Fruit{
    public void printDetails(){
        System.out.println("This is a fruit");
    }
}
class Apple extends Fruit{
    public void printDetails(){
        System.out.print (" This is an apple");
    }
    public void nutrition (){
        System.out.print ("Calories,Protein,Carb etc");
    }
}
public class Main{
    public static void main(String args[]){
        Fruit fruit1 = new Fruit();
        Fruit fruit2 = new Apple();
        fruit1.nutrition();
        fruit2.printDetails();
    }
}
```

Select one:

- This is a fruit This is an apple
- This is a fruit
- Calories,Protein,Carb etc This is an apple
- Compilation error
- Calories,Protein,Carb etc

Select the correct output for the following program.

```
CLASS A{  
    int min;  
    int max;  
    public A(int a, int b){  
        this.min = a;  
        this.max = b;  
    }  
  
    public void updateMinMax(A a){  
        if(a.min < this.min)  
            this.min = a.min;  
        if(a.max > this.max)  
            this.max = a.max;  
    }  
    public void printMinMax(){  
        System.out.print(this.min + ":" + this.max + ",");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        A a1 = new A(2,4);  
        A a2 = new A(1,3);  
  
        a1.updateMinMax(a2);  
        a2.updateMinMax(a1);  
  
        a1.printMinMax();  
        a2.printMinMax();  
    }  
}
```

Select one:

- 1:4,1:4,
- 2:4,1:3
- 1:3,1:3,
- 1:3,2:4
- 2:3,1:4,

 NetExam
Sri Lanka Institute of Information Technology

When an overridden method is called from within a subclass, it will always refer to the version of that method defined by the

Select one:

a. Super class

b. Interpreter will choose randomly

c. None of the mention

d. Compiler will choose randomly

e. Subclass

Moodle → X C ⓘ ↻ 12/07/2018 09:48

 NetExam
Sri Lanka Institute of Information Technology

Section 8
1 question answered
1 question unanswered
Total 1 question

Drag and drop the most suitable word.

When override the methods the return types of both methods should be same or the overridden method return type should be **a sub class** of the return type of superclass method. And methods signature should be same. Access modifiers of the method should be **same or wider**.

a sub class a super class Object class

same or wider same lower

Quiz navigation
Finish attempt...
Time left 0:36:38
MCQ QUESTION
1 2 3
9 10 11
17 18 19
STRUCTURE
21 22

Moodle

→ X C | i | ↗

NetExam
Sri Lanka Institute of Information Technology

Section 8
1 out of 1
Flag question

Drag and drop the most suitable word.

When override the methods the return types of both methods should be same or the overridden method return type should be of the return type of superclass method. And methods signature should be same. Access modifiers of the method should be .

a sub class a super class Object class

same or wider same lower

Next page

Quiz navigation

Finish attempt...
Time left 0:36:38

MCQ QUESTION

1	2	3
9	10	11
17	18	19

STRUCTURE

31	32
----	----

Select the correct output for the following program.

```
class Demo{  
    int x;  
    int y;  
    public Demo(int a, int b){  
        this.x = a;  
        this.y = b;  
    }  
    public Demo nextDemo(){  
        Demo c = new Demo(this.a*2), (this.b+2));  
        return c;  
    }  
    public void printValues(){  
        System.out.print(this.a + ":" + this.b + ",");  
    }  
}  
public class Main{  
    public static void main(String args[]){  
        Demo d1 = new Demo(2,4);  
        Demo d2;  
        d2 = d1.nextDemo();  
        d2.printValues();  
        d1.printValues();  
    }  
}
```

Select one:

- 2:4;4;8,
- 4;6;2;4
- 4;6;4;6,
- 2;4;4;6,
- 2;4;2;4,

NetExam
Sri Lanka Institute of Information Technology

8 answered out of question

Select the correct method signatures to implement cloning objects of following class Patient.

```
class Patient{  
    int age;  
    String name;  
    public Patient(){  
        this.age = 0;  
        this.name = "No name";  
    }  
    public Patient (int age, String name) {  
        this.age = age;  
        this.name = name;  
    }  
}
```

Select one:

- Implement public void clone(Patient p)
- Implement public void Patient(Patient p)
- Implement public Patient(String p)
- Implement public Patient(Patient p)
- Implement public void clonePatient(int age, String name)

ASUS V

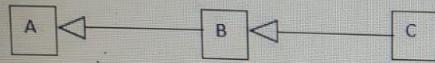
1 2 3 4 5 6 7

Q W E R T V

on 5

not answered
out of
question

Classes A, B and C are having the inheritance relationship as given below.



If public void sumof2() is a method signature of a method in class B, select the correct way to implement method overriding.

Select one:-

- Implement public int sumof2() in class C
- Implement public void sumof2() in class C
- Implement public void sumof2() in class C
- Implement public void sumof2() in class B
- Implement public void sumof2() in class A

Next page

Examine the below classes.

```
1  public class Item {  
2      double capacity;  
3      double height;  
4      public Item() {  
5          }  
6          }  
7          public Item(double capacity, double height) {  
8              super();  
9              System.out.println("Item constructor");  
10             this.capacity = capacity;  
11             this.height = height;  
12         }  
13     }  
14 }  
15 }  
16 class Bottle extends Item{  
17     boolean lid;  
18  
19     public Bottle(double capacity, double height, boolean lid) {  
20         this(capacity, height);  
21         System.out.println("Bottle constructor 1");  
22         this.lid = lid;  
23     }  
24     public Bottle(double capacity, double height) {  
25         System.out.println("Bottle constructor 2");  
26     }  
27 }  
28 }  
29 class Can extends Bottle{  
30     String type;  
31  
32     public Can(double capacity, double height, boolean lid, String type) {  
33         super(capacity, height, lid);  
34         System.out.println("Can constructor");  
35         this.type = type;  
36     }  
37 }  
38 }
```

Can ob=new Can(25.0, 30, false, "Double");

Select one:

- item constructor
- Bottle constructor 1
- Can constructor

Given the following interface and class defined in a file named Traceable.java , what is the result of compiling this code?

```
1. public interface Traceable {  
2.     public static int MAX_TRACE;  
3.     public void trace();  
4. }  
5.  
6. class Picture implements Traceable {  
7.     public void trace() {  
8.         System.out.println("Tracing a picture");  
9.     }  
10. }
```

Select one:

- One bytecode file: Traceable.class
- Compiler error on line 2 MAX_TRACE variable should be initialized
- Compiler error on line 6
- Compiler error on line 3
- Two bytecode files: Traceable.class and Picture.class

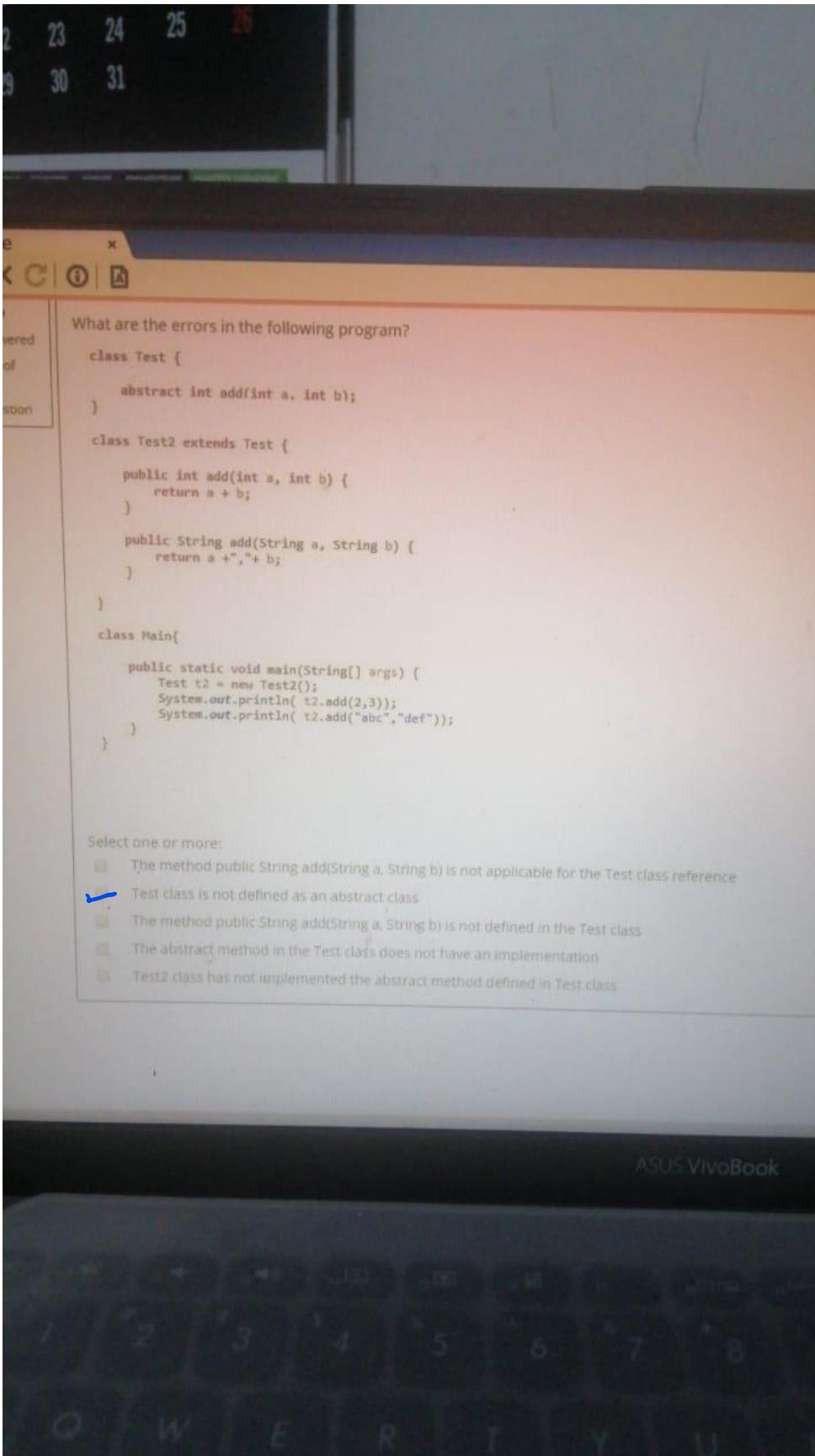
ASUS VivoBook

fill the blank with the correct word

When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer:

[Next page](#)





Refer the bellow code and choose the cases where line D will be executed

```
//Line A
try {
    // Line B
} catch (SomeException se) {
    // Line C
} finally {
    // Line D
}
```

Select one or more:

- When line A throws an exception **program will terminate**
- When Line B throws any type of exception
- When Line B executed successfully
- When Line B throws an exception of type SomeException
- When Line B throws an exception of sub class of SomeException

5

swered
t of
question

Consider the following statements to import the Scanner class in order to read a keyboard input.

- A) import java.util.Scanner;
- B) import java.util;
- C) import java.util.*;

Select one:

- All three methods are correct
- Only A is correct
- A and C are correct
- B and C are Correct
- A and B are Correct

```
abstract class Test {  
    abstract int add(int a, int b, int c);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "," + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- The main method is static and we cannot call a non-static method from a static method
- The method public String add(String a, String b) is not applicable for the Test class reference
- Test2 class has not implemented the abstract method abstract int add(int a, int b, int c);
- The method public int add(int a, int b) is not applicable for the Test class reference
- We cannot create a Test2 object and assign it to a Test class reference.

ASUS VivoBo



Moodle

Question 15
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) {  
        C1 c1 = new C1();  
        c1.method1(2, 3);  
    }  
  
    abstract class A {  
        abstract void method1(int a, int b);  
        abstract void method2(String a);  
    }  
  
    abstract class C1 implements A {  
        public void method1(int a, int b) {  
        }  
    }  
}
```

If a child class did not override all the abstract method of the parent, then the child class also become abstract.

Select one or more:

- Defining C1 as abstract
- Creating a new object from C1
- Usage of implements keyword
- Defining A as an abstract class
- Defining Methods in A as abstract

Cannot create instances (object) from an abstract class.

Question 19

Not yet answered
Marked out of
1.00

Flag question

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(1, "cc");
System.out.println(sb);
1 4 append Insert
4 append 1 insert
Insert append 1 4
```

What is the **correct** statement of the following?

Select one:

these two have conflict meaning

- You can use both abstract and final keywords together when you create a method
- When you declare a method in the interface those methods are by default abstract and final by default it is public & abstract
- You can use all four access modifiers when you create a class only public & protected can be used
- None of the mentioned
- You can use abstract key word for a method, class and variables in Java

SLIIT
Sri Lanka Institute of Information Technology

Question 19
Not yet answered
Marked out of 1.00
 Flag question

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- An exception is thrown on line 6.
- Line 1 generates a compiler error.
- Line 4 generates a compiler error.
- equal
- not equal

tion 10

yet answered

ked out of

Flag question

What is the result of the following code?

```
1. public class Shape {  
2.     private String color;  
3.  
4.     public Shape(String color) {  
5.         System.out.print("Shape");  
6.         this.color = color;  
7.     }  
8.  
9.     public static void main(String [] args) {  
10.        new Rectangle();  
11.    }  
12. }  
13.  
14. class Rectangle extends Shape {  
15.     public Rectangle() {  
16.         System.out.print("Rectangle");  
17.     }  
18. }
```

What can be select as true statement/s?

Select one:

- ShapeRectangle
- Rectangle
- RectangleShape
- Line 4 generates a compiler error
- Line 15 generates a compilation error

✓ line 15 generates a compilation error

NetExam
Sri Lanka Institute of Information Technology

Select the method/s which are incorrectly declared inside the Writer class.

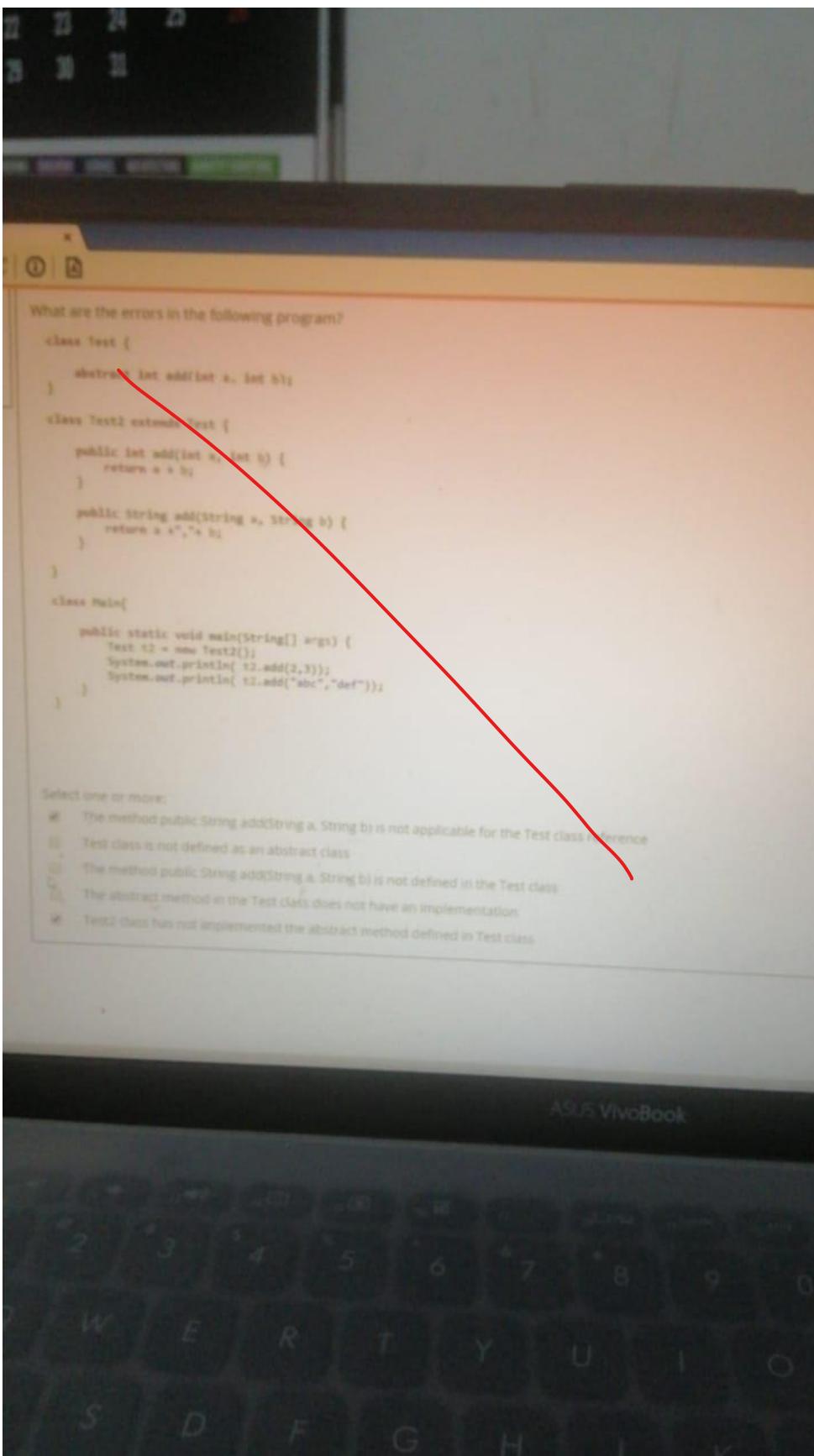
```
public abstract class Writer {  
    X public void method1(); need an implementation  
    public final void method2() {};  
    public static void method3() {};  
    X public abstract static void method4();  
    X public abstract final void method5();  
}
```

Select one or more:

- method1
- method5
- method2
- method4
- method3

DELL





Question 20

Not yet answered

Marked out of

1.00

Flag question

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- Line 4 generates a compiler error.
- not equal
- equal
- An exception is thrown on line 6.
- Line 1 generates a compiler error.

Next page

Refer the bellow code and choose the cases where line D will be executed ?

```
//Line A  
try {  
    // Line B  
} catch (SomeException se) {  
    // Line C  
} finally {  
    // Line D  
}
```

- Select one or more:
- When line A throws an exception
 - When Line B executed successfully
 - When Line B throws an exception of type SomeException
 - When Line B throws any type of exception
 - When Line B throws an exception of sub class of SomeException

```
class Main {  
    public static void main(String[] args) {  
        C1 c1 = new C1();  
        c1.method1(2, 3);  
    }  
  
    abstract class A {  
  
        abstract void method1(int a, int b);  
        abstract void method2(String a);  
    }  
  
    abstract class C1 implements A {  
  
        public void method1(int a, int b) {  
        }  
    }  
}
```

Select one or more:

- Creating a new object from C1
- Defining Methods in A as abstract
- Defining C1 as abstract
- Defining A as an abstract class
- Usage of implements keyword

```
16         System.out.println("E");
17     }
18     finally {
19         System.out.println("F");
20     }
21     System.out.println("G");
22 }
23
24 }
```

Select one:

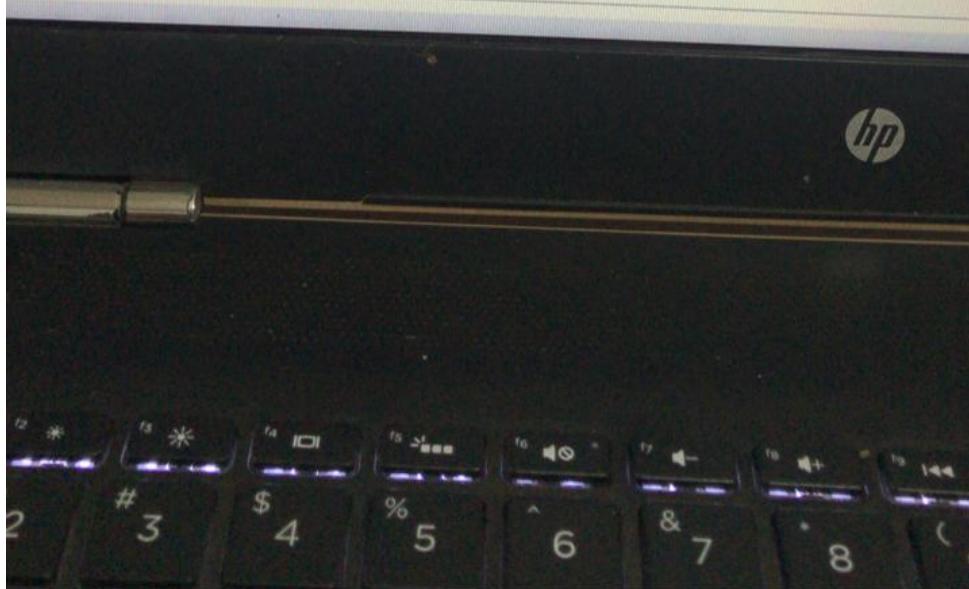
E
 F

B
 F
 G

B
 F

E
 F
 G

C
 F



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 s1 = new S1(); is wrong as we cannot create an object from Class S1

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

Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong

S2 s2 = new S2(); is wrong as we cannot create an object from class S2

*



≡ Quiz n

Finish attempt
Time left 0:1

MCQ QUES

1	2
8	9
15	16

STRUCTURE

21	22
----	----

FEEDBACK

23

Select the correct Exception output of the following program.

```
public static void main(String[] args) {
    try{
        try{
            int value = 10/0;
        }catch (ArithmaticException e) {
            System.out.println("ArithmaticException");
        }finally{
            System.out.println("Inner finally");
            int marks [] = new int[] {10};
            int value = marks[1];
        }
    }catch (NumberFormatException e) {
        System.out.println("NumberFormatException");
    }catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("ArrayIndexOutOfBoundsException");
    }finally {
        System.out.println("Outer finally");
    }
}
```

Select one:

- ArithmeticException
Inner finally
Outer finally
- Line unreachable runtime error
- ArithmeticException
Inner finally
ArrayIndexOutOfBoundsException
Outer finally
- Compile Error
- ArithmeticException
ArrayIndexOutOfBoundsException
Outer finally

Select the method/s which are incorrectly declared inside the Writer class.

```
public abstract class Writer {  
    public void method1();  
    public final void method2() {};  
    public static void method3() {};  
    public abstract static void method4();  
    public abstract final void method5();  
}
```

Select one or more:

- method1
- method5
- method2
- method4
- method3

DELL



Moodle

Question 16
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) {  
        C1 c1 = new C1();  
        c1.method3(2, 3);  
    }  
}  
  
interface I1 {  
    void method1(int a, int b);  
    void method2(String a);  
}  
  
interface I2 {  
    void method3(int a, int b);  
    void method2(String a);  
}  
  
class C1 extends I1, I2 {  
    public void method1(int a, int b) {}  
    public void method2(String a) {}  
    public void method3(int a, int b) {}  
}
```

Select one or more:

- C1 not implementing the method 2 twice
- Using the extends key word for inheritance of interfaces
- Both interfaces having the same method void method2(String a);
- Creating a new C1 object
- Inheriting from multiple interfaces

1
swered
ut of
uestion

What is the **correct** statement of the following?

Select one:

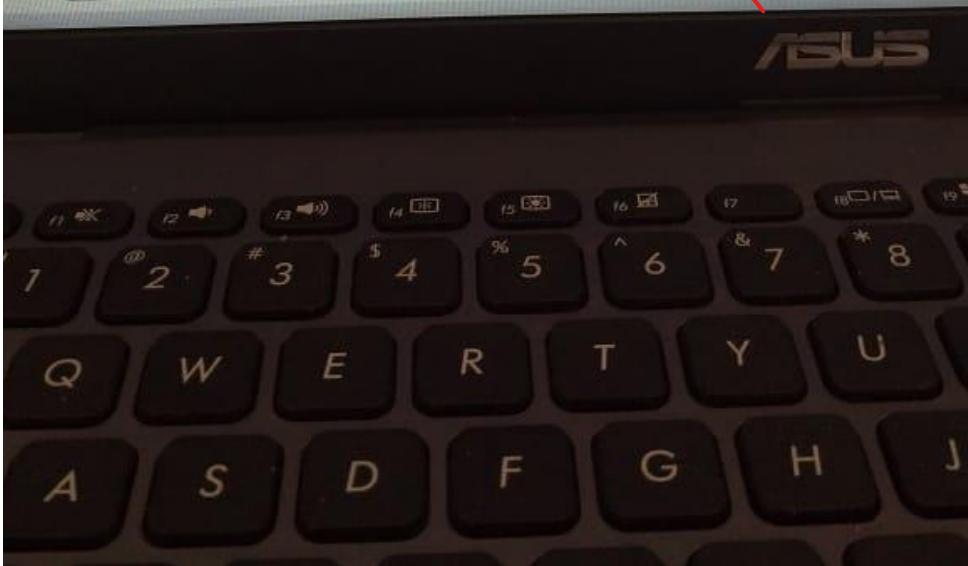
- None of the mentioned
- When you declare a method in the interface those methods are by default abstract and final
- You can use abstract key word for a method, class and variables in Java
- You can use both abstract and final keywords together when you create a method
- You can use all four access modifiers when you create a class

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

```
class Main {  
    public static void main(String[] args) {  
        C1 c1 = new C1();  
        c1.method3(2, 3);  
    }  
}  
  
interface I1 {  
    void method1(int a, int b);  
    void method2(String a);  
}  
  
interface I2 {  
    void method3(int a, int b);  
    void method2(String a);  
}  
  
class C1 extends I1, I2 {  
    public void method1(int a, int b) {  
    }  
    public void method2(String a) {  
    }  
    public void method3(int a, int b) {  
    }  
}
```

Select one or more:

- C1 not implementing the method 2 twice
- Inheriting from multiple interfaces
- Both interfaces having the same method void method2(String a)
- Using the extends key word for inheritance of interfaces
- Creating a new C1 object



What are the errors in the following program?

```
class Test {  
    abstract int add(int a, int b);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "+" + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- The method public String add(String a, String b) is not applicable for the Test class reference
- Test class is not defined as an abstract class
- The abstract method in the Test class does not have an implementation
- Test2 class has not implemented the abstract method defined in Test class
- The method public String add(String a, String b) is not defined in the Test class



 NetExam
Sri Lanka Institute of Information Technology

fill the blank with the correct word
When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer: **override**

[Next page](#)

Question 15
Not yet answered
Marked out of 1.00
[Flag question](#)

What is the output of below program?

```
abstract class Tester {  
    public static void test() {  
        System.out.println("Testing...");  
    }  
}  
class laboratorian extends Tester {  
    public static void test() {  
        System.out.println("Testing blood");  
    }  
}  
public class Programmer extends Tester {  
    public static void test() {  
        System.out.println("Testing code");  
    }  
}  
public static void main(String[] args) {  
    Tester ob = new Programmer();  
    ob.test();  
}
```

Static methods are bound to reference type not the object type
Above code although the object type is programmer the test function is called based on the reference type Tester.

Select one:

- Testing... Testing...
- Compilation Error
- An exception
- Testing blood
- Testing code

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

Institute of Information Technology

What is the **correct** statement of the following?

Select one:

- You can use all four access modifiers when you create a class
- When you declare a method in the interface those methods are by default abstract and final
- None of the mentioned
- You can use both abstract and final keywords together when you create a method
- You can use abstract key word for a method, class and variables in Java

Next page

Question 18
Not yet answered
Marked out of 1.00
Flag question

Consider the below code:

```
public class ColorException extends Exception {}  
public class White {  
    public String printLine(String color) throws ColorException {  
        if(color == null) { throw new ColorException(); }  
        return "Hello " + color;  
    }  
}
```

You are adding below code in your application:

Line A : White white = new White();
Line B: System.out.println(white.printLine ("Ceramic"));
Select true answers from below:

Select one or more:

- Line B will compile if the enclosing method throws a ColorException.
- Line A can throw the unchecked exception ColorException
- Class White will not compile
- Line B can throw the unchecked exception ColorException
- Line B will compile if enclosed in a try block, where ColorException is caught.

this is a checked exception bcz it extends from Exception class

Quiz navigation
Finish attempt
Time left 0:21:1
MCQ QUESTIONS
1 2 3
8 9 10
15 16 17
STRUCTURED
21 22
FEEDBACK
23

 NetExam
Sri Lanka Institute of Information Technology

Select the correct output of the following program.

```
public static void main(String[] args) {
    int [] arr = new int[]{65,66,67,68};
    char [] charArr = new char[arr.length];
    for (int i = 0; i < arr.length; i++) {
        charArr[i] = (char)arr[i];
    }
    for (char c : charArr) {
        System.out.print(c + ", ");
    }
}
```

charArr[0] = 65
charArr[1] = 66
charArr[2] = 67
charArr[3] = 68

ASCII values

65 => A
66 => B
67 => C
68 => D

output :
A,B,C,D,

Select one:

A, B, C, D.
 A, B, C, D
 65, 66, 67, 68
 65, 66, 67, 68,
 Runtime Error mismatch of int [] to char[]

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
- Defining and implementing the non abstract method String add(int a, String b) in class S1 is wrong
- 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
- Cannot use s2.add("abc", "def") as that method is not implemented in class S2

Time left 0:1

MCQ QUES

STRUCTURE

FEEDBACK

What is the output of the following code?

```
class Calculator
{
    public int val;
    private int val2;
    void pass(int res, int res1)
    {
        val = res + 1;
        val2 = res1;
    }
    void print()
    {
        System.out.println(" " + res1);   res1 is a parameter
    }
}

public class accessSpecifier
{
    public static void main(String args[])
    {
        Calculator cal = new Calculator();
        cal.pass(2, 3);
        System.out.println(cal.res);   res is a parameter
        cal.print();
    }
}
```

val = 2+1 =3
val2 = 3

res1 is a parameter

res is a parameter

Select one:

- null
- Runtime error
- 2,3
- 3,3
- Compilation error

Moodle

NetExam

Sri Lanka Institute of Information Technology

on 15
answered
out of
g question

fill the blank with the correct word
When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer: override

Next page

NetExam

Sri Lanka Institute of Information Technology

Select the most suitable word from the drop down.

When you create any class in the interface, are considered as final and you must override it. Inside the override concrete class when you implements and interface you do not need to override those methods.

✓ concrete class is a class that provide implementations for all of its methods and can be instantiated (can create objects)

Next page

Consider the given code

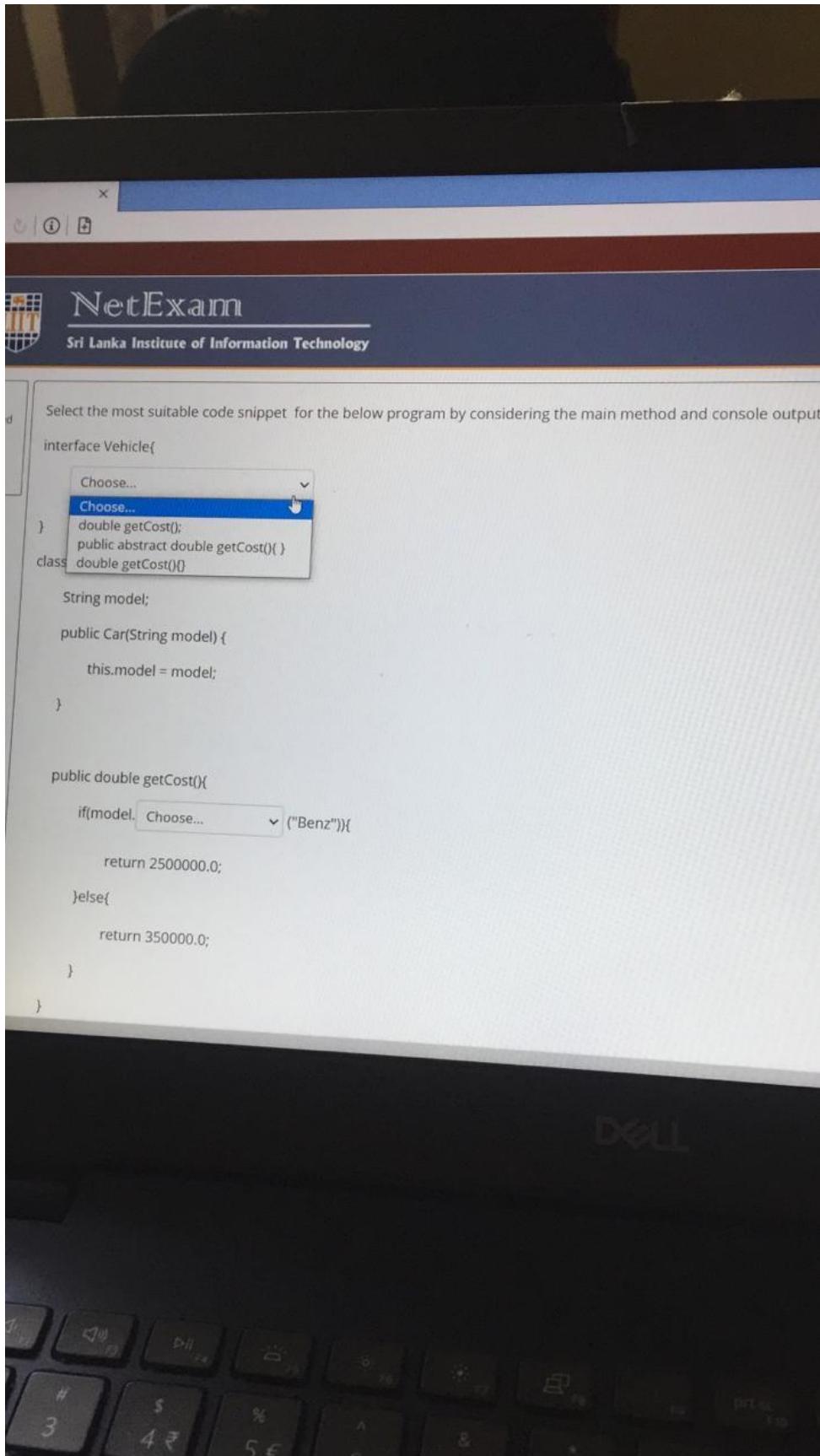
```
4. public class Frodo extends Hobbit {  
5.     public static void main(String[] args) {  
6.         Short myGold = 7;    short  
7.         System.out.println(countGold(myGold, 6));  parameter mismatch  
8.     }  
9. }  
10. class Hobbit {  
11.     int countGold(int x, int y) { return x + y; }  
12. }
```

What is the result?

Select one:

- Compilation fails due to an error on line 11
- 13
- Compilation fails due to an error on line 7
- Compilation fails due to multiple errors
- Compilation fails due to an error on line 6

Next page



Moodle

Not yet answered
Marked out of 1.00
Flag question

```
interface Vehicle{
    public abstract double getCost();
}
class Car implements Vehicle {
    String model;
    public Car(String model) {
        this.model = model;
    }
    public double getCost() {
        if(model.equalsIgnoreCase("Benz")){
            return 2500000.0;
        }else{
            return 350000.0;
        }
    }
}
```

Finish attempt
Time left 0:19:1

MCQ QUESTION

STRUCTURED QUESTIONS

FEEDBACK

Question 13

Not yet answered
Marked out of 1.00
Flag question

What are the errors in the following program?

```
class Test {
    abstract int add(int a, int b);
}

class Test2 extends Test {
    public int add(int a, int b) {
        return a + b;
    }

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

class Main{
    public static void main(String[] args) {
        Test t2 = new Test2();
        System.out.println( t2.add(2,3));
        System.out.println( t2.add("abc", "def"));
    }
}
```

Quiz navigation

Finish attempt
Time left 0:19:1

MCQ QUESTION

STRUCTURED QUESTIONS

FEEDBACK

Select one or more:

- Test class is not defined as an abstract class
- Test2 class has not implemented the abstract method defined in Test class
- The method public String add(String a, String b) is not defined in the Test class
- The abstract method in the Test class does not have an implementation
- The method public String add(String a, String b) is not applicable for the Test class reference

Given the following definitions:

1. //Readable.java
 2. public interface Readable {
 3. public abstract void read();
 4. }
-
1. //SpellCheck.java
 2. public interface SpellCheck extends Readable {
 3. public void checkSpelling();
 4. }

which of the following statements are true?

Select one:

- An Interface cannot extend another interface.
- The SpellCheck interface does not compile
- A class that implements SpellCheck inherits both the checkSpelling and read methods.
- A class that implements Readable must override the read method.
- A class that implements SpellCheck only inherits the checkSpelling method.

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

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

class Main{
    public static void main(String[] args) {
        Test t2 = new Test();
        System.out.println(t2.add(2,3));
        System.out.println(t2.add("abc","def"));
    }
}
```

Select one or more:

- The method public String add(String a, String b) is not applicable for the Test class reference
- Test2 class has not implemented the abstract method abstract int add(int a, int b, int c);
- We cannot create a Test2 object and assign it to a Test class reference.
- The main method is static and we cannot call a non-static method from a static method
- The method public int add(int a, int b) is not applicable for the Test class reference

```
public static void main(String[] args) {
    System.out.println("Enter sizes of the Square & Traingle");
    Scanner scanner = new Scanner(System.in);
    int size = scanner.nextInt();
    for [ ] {
        for [ ] {
            System.out.print("* ");
        }
        for [ ] {
            System.out.print(" ");
        }
        for (int y = 1; y <= x; y++) {
            System.out.print("* ");
        }
        System.out.println();
    }
}

(int x = 1; x <= size; x++) (int x = 0; x < size; x++) (int x = 0; x <= size; x++)
(int y = size; y >= x; y--) (int y = 0; y < x; y++) (int y = 0; y <= x; y++)
(int y = 1; y <= x; y++) (int y = 0; y <= x; y++) (int y = 0; y
```

You submitted

bad test of

Flag question

```
class Test {
    abstract int add(int a, int b);
}

class Test2 extends Test {
    public int add(int a, int b) {
        return a + b;
    }

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

class Main{
    public static void main(String[] args) {
        Test t2 = new Test2();
        System.out.println( t2.add(2,3));
        System.out.println( t2.add("abc","def"));
    }
}
```

Select one or more:

- The abstract method in the Test class does not have an implementation
- Test2 class has not implemented the abstract method defined in Test class
- The method public String add(String a, String b) is not defined in the Test class
- The method public String add(String a, String b) is not applicable for the Test class reference
- Test class is not defined as an abstract class

**2**swered
ut of
uestion

Consider the given code

```
4. public class Frodo extends Hobbit {  
5.     public static void main(String[] args) {  
6.         Short myGold = 7;  
7.         System.out.println(countGold(myGold, 6));  
8.     }  
9. }  
10. class Hobbit {  
11.     int countGold(int x, int y) { return x + y; }  
12. }
```

What is the result?

Select one:

- Compilation fails due to an error on line 6
- Compilation fails due to an error on line 7
- Compilation fails due to multiple errors
- 13
- Compilation fails due to an error on line 11

[Next page](#)

on 16

it answered

out of

g question

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

```
class Main {  
    public static void main(String[] args) {  
        C1 c1 = new C1();  
        c1.method1(2, 3);  
    }  
}  
  
abstract class A {  
    abstract void method1(int a, int b);  
    abstract void method2(String a);  
}  
  
abstract class C1 implements A {  
    public void method1(int a, int b) {  
    }  
}
```

Select one or more:

- Defining A as an abstract class
- Defining C1 as abstract
- Creating a new object from C1
- Usage of implements keyword
- Defining Methods in A as abstract

Question 12

Not yet answered

Marked out of
1.00[Flag question](#)

Select the correct output of the following program.

```
public static void main(String[] args) {  
    int [] arr = new int[]{65,66,67,68};  
    char [] charArr = new char[arr.length];  
    for (int i = 0; i < arr.length; i++) {  
        charArr[i] = (char)arr[i];  
    }  
    for (char c : charArr) {  
        System.out.print(c + ", ");  
    }  
}
```

Select one:

- 65, 66, 67, 68
- A, B, C, D
- A, B, C, D,
- 65, 66, 67, 68,
- Runtime Error mismatch of int [] to char[]

[Next page](#)**NetExam**

Sri Lanka Institute of Information Technology

Question 17

Not yet answered

Marked out of
1.00[Flag question](#)

What will be the output of below code?

```
static void test() {  
    try {  
        // code creating and declaring a string variable to null  
        // calling toString() method for the string variable and printing  
    }  
    finally {  
        System.out.print("finally ");  
    }  
}  
public static void main(String[] args) {  
    try {  
        test();  
    }  
    catch (Exception ex) {  
        System.out.print("exception ");  
    }  
}
```

Select one:

- a. Compilation fails
- b. Finally exception
- c. Finally
- d. Null
- e. Null finally

[Finish at](#)

Time left

1

MCQ QUEST

1 2

9 10

17 18

STRUCTURED

21 22

23

FEEDBACK

DELL

Session 16
Not yet answered
Marked out of
1.00
Flag question

What is the output of below program?

```
abstract class Tester {  
    public static void test() {  
        System.out.println("Testing...");  
    }  
}  
class laboratorian extends Tester {  
    public static void test() {  
        System.out.println("Testing blood");  
    }  
}  
public class Programmer extends Tester {  
    public static void test() {  
        System.out.println("Testing code");  
    }  
}  
public static void main(String[] args) {  
    Tester ob = new Programmer();  
    ob.test();  
}
```

Select one:

- Testing...
- Testing code
- Testing blood
- Compilation Error
- An exception



Sri Lanka Institute of Information Technology

5
Covered
of
session

Consider the following statements to import the Scanner class in order to read a keyboard input.

- A) import java.util.Scanner;
- B) import java.util;
- C) import java.util.*;

Select one:

- All three methods are correct
- A and C are correct
- Only A is correct
- B and C are Correct
- A and B are Correct

```
CLASS MAIN {
    public static void main(String[] args) {
        C1 c1 = new C1();
        c1.method3(2, 3);
    }
}

interface I1 {
    void method1(int a, int b);
    void method2(String a);
}

interface I2 {
    void method3(int a, int b);
    void method2(String a);
}

class C1 extends I1, I2 {
    public void method1(int a, int b) {
    }

    public void method2(String a) {
    }

    public void method3(int a, int b) {
    }
}
```

Select one or more:

- Creating a new C1 object
- Both interfaces having the same method void method2(String a);
- Inheriting from multiple interfaces
- C1 not implementing the method 2 twice
- Using the extends key word for inheritance of interfaces

fill the blank with the correct word

When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer:

[Next page](#)

2 23 24 25 26
9 30 31

e x

< C | ① | A

What are the errors in the following program?

```
class Test {  
    abstract int add(int a, int b);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "," + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3)  
        System.out.println( t2.add("abc", "def"));  
    }  
}
```

Select one or more:

- The method public String add(String a, String b) is not applicable for the Test class reference
- Test class is not defined as an abstract class
- The method public String add(String a, String b) is not defined in the Test class
- The abstract method in the Test class does not have an implementation
- Test2 class has not implemented the abstract method defined in Test class

ASUS VivoBook

1 2 3 4 5 6 7 8
Q W E R T Y U



Refer the bellow code and choose the cases where line D will be executed

```
//Line A
try {
    // Line B
} catch (SomeException se) {
    // Line C
} finally {
    // Line D
}
```

Select one or more:

- When line A throws an exception
- When Line B throws any type of exception
- When Line B executed successfully
- When Line B throws an exception of type SomeException
- When Line B throws an exception of sub class of SomeException

5

swered
t of
question

Consider the following statements to import the Scanner class in order to read a keyboard input.

- A) import java.util.Scanner;
- B) import java.util;
- C) import java.util.*;

Select one:

- All three methods are correct
- Only A is correct
- A and C are correct
- B and C are Correct
- A and B are Correct

```
abstract class Test {  
    abstract int add(int a, int b, int c);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "," + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- The main method is static and we cannot call a non-static method from a static method
- The method public String add(String a, String b) is not applicable for the Test class reference
- Test2 class has not implemented the abstract method abstract int add(int a, int b, int c);
- The method public int add(int a, int b) is not applicable for the Test class reference
- We cannot create a Test2 object and assign it to a Test class reference.

ASUS VivoBo



Moodle

Question 15

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) {  
        C1 c1 = new C1();  
        c1.method1(2, 3);  
    }  
  
    abstract class A {  
        abstract void method1(int a, int b);  
        abstract void method2(String a);  
    }  
  
    abstract class C1 implements A {  
        public void method1(int a, int b) {  
        }  
    }  
}
```

Select one or more:

- Defining C1 as abstract
- Creating a new object from C1
- Usage of implements keyword
- Defining A as an abstract class
- Defining Methods in A as abstract

Question 19

Not yet answered
Marked out of
1.00

Flag question

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

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

What is the **correct** statement of the following?

Select one:

- You can use both abstract and final keywords together when you create a method
- When you declare a method in the interface those methods are by default abstract and final
- You can use all four access modifiers when you create a class
- None of the mentioned
- You can use abstract key word for a method, class and variables in Java

SLIIT
Sri Lanka Institute of Information Technology

Question 19
Not yet answered
Marked out of 1.00
 Flag question

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- An exception is thrown on line 6.
- Line 1 generates a compiler error.
- Line 4 generates a compiler error.
- equal
- not equal

tion 10

yet answered

ked out of

Flag question

What is the result of the following code?

```
1. public class Shape {  
2.     private String color;  
3.  
4.     public Shape(String color) {  
5.         System.out.print("Shape");  
6.         this.color = color;  
7.     }  
8.  
9.     public static void main(String [] args) {  
10.        new Rectangle();  
11.    }  
12. }  
13.  
14. class Rectangle extends Shape {  
15.     public Rectangle() {  
16.         System.out.print("Rectangle");  
17.     }  
18. }
```

What can be select as true statement/s?

Select one:

- ShapeRectangle
- Rectangle
- RectangleShape
- Line 4 generates a compiler error
- Line 15 generates a compiler error ✓

NetExam
Sri Lanka Institute of Information Technology

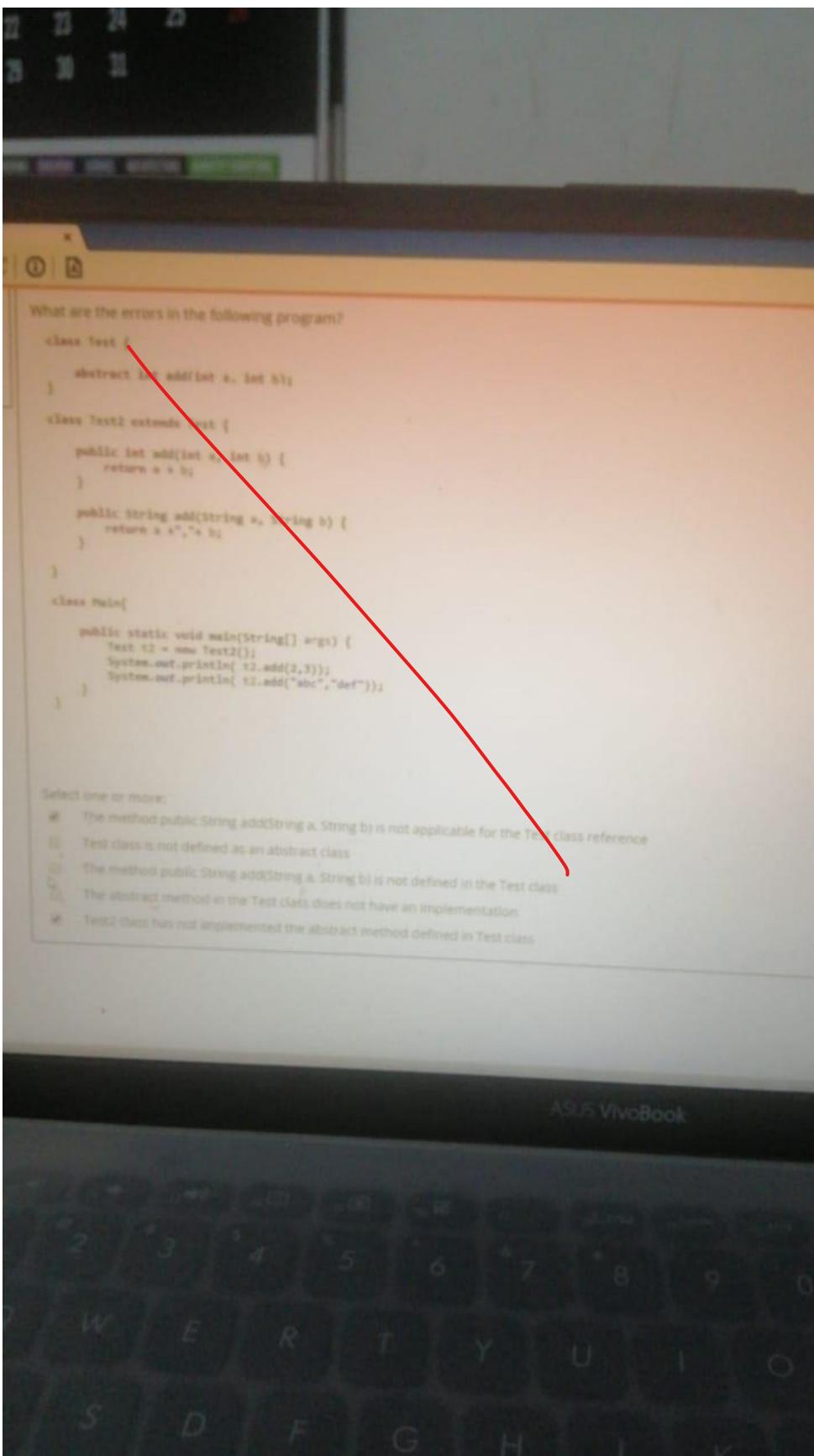
Select the method/s which are incorrectly declared inside the Writer class.

```
public abstract class Writer {  
    public void method1();  
    public final void method2() {};  
    public static void method3() {};  
    public abstract static void method4();  
    public abstract final void method5();  
}
```

Select one or more:

- method1
- method5
- method2
- method4
- method3

DELL



Question 20

Not yet answered

Marked out of

1.00

Flag question

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- Line 4 generates a compiler error.
- not equal
- equal
- An exception is thrown on line 6.
- Line 1 generates a compiler error.

Next page

Refer the bellow code and choose the cases where line D will be executed ?

```
//Line A  
try {  
    // Line B  
} catch (SomeException se) {  
    // Line C  
    finally {  
        // Line D  
    }  
}
```

- Select one or more:
- When line A throws an exception
 - When Line B executed successfully
 - When Line B throws an exception of type SomeException
 - When Line B throws any type of exception
 - When Line B throws an exception of sub class of SomeException

```
class Main {  
    public static void main(String[] args) {  
        C1 c1 = new C1();  
        c1.method1(2, 3);  
    }  
  
    abstract class A {  
  
        abstract void method1(int a, int b);  
        abstract void method2(String a);  
    }  
  
    abstract class C1 implements A {  
  
        public void method1(int a, int b) {  
        }  
    }  
}
```

Select one or more:

- Creating a new object from C1
- Defining Methods in A as abstract
- Defining C1 as abstract
- Defining A as an abstract class
- Usage of implements keyword

```
16         System.out.println("E");
17     }
18     finally {
19         System.out.println("F");
20     }
21     System.out.println("G");
22 }
23
24 }
```

Select one:

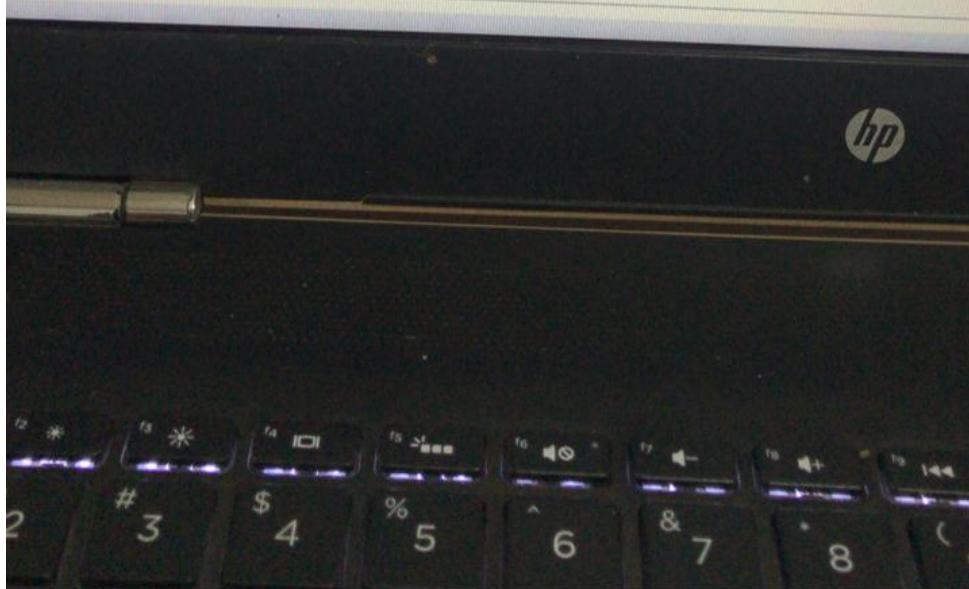
E
 F

B
 F
 G

B
 F

E
 F
 G

C
 F



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 s1 = new S1(); is wrong as we cannot create an object from Class S1

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

Defining and implementing the non abstract method String add(String a, String b) in class S1 is wrong

S2 s2 = new S2(); is wrong as we cannot create an object from class S2

•

Finish attempt
Time left 0:1

MCQ QUES
1 2
8 9
15 16

STRUCTURE
21 22

FEEDBACK
23

Select the correct Exception output of the following program.

```
public static void main(String[] args) {
    try{
        try{
            int value = 10/0;
        }catch (ArithmaticException e) {
            System.out.println("ArithmaticException");
        }finally{
            System.out.println("Inner finally");
            int marks [] = new int[] {10};
            int value = marks[1];
        }
    }catch (NumberFormatException e) {
        System.out.println("NumberFormatException");
    }catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("ArrayIndexOutOfBoundsException");
    }finally {
        System.out.println("Outer finally");
    }
}
```

Select one:

- ArithmeticException
Inner finally
Outer finally
- Line unreachable runtime error
- ArithmeticException
Inner finally
ArrayIndexOutOfBoundsException
Outer finally
- Compile Error
- ArithmeticException
ArrayIndexOutOfBoundsException
Outer finally

Select the method/s which are incorrectly declared inside the Writer class.

```
public abstract class Writer {  
    public void method1();  
    public final void method2() {};  
    public static void method3() {};  
    public abstract static void method4();  
    public abstract final void method5();  
}
```

Select one or more:

- method1
- method5
- method2
- method4
- method3

DELL



Moodle

Question 16

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) {  
        C1 c1 = new C1();  
        c1.method3(2, 3);  
    }  
}  
  
interface I1 {  
    void method1(int a, int b);  
    void method2(String a);  
}  
  
interface I2 {  
    void method3(int a, int b);  
    void method2(String a);  
}  
  
class C1 extends I1, I2 {  
    public void method1(int a, int b) {}  
    public void method2(String a) {}  
    public void method3(int a, int b) {}  
}
```

Select one or more:

- C1 not implementing the method 2 twice
- Using the extends key word for inheritance of interfaces
- Both interfaces having the same method void method2(String a);
- Creating a new C1 object
- Inheriting from multiple interfaces

1
swered
ut of
uestion

What is the **correct** statement of the following?

Select one:

- None of the mentioned
- When you declare a method in the interface those methods are by default abstract and final
- You can use abstract key word for a method, class and variables in Java
- You can use both abstract and final keywords together when you create a method
- You can use all four access modifiers when you create a class

Ne

x

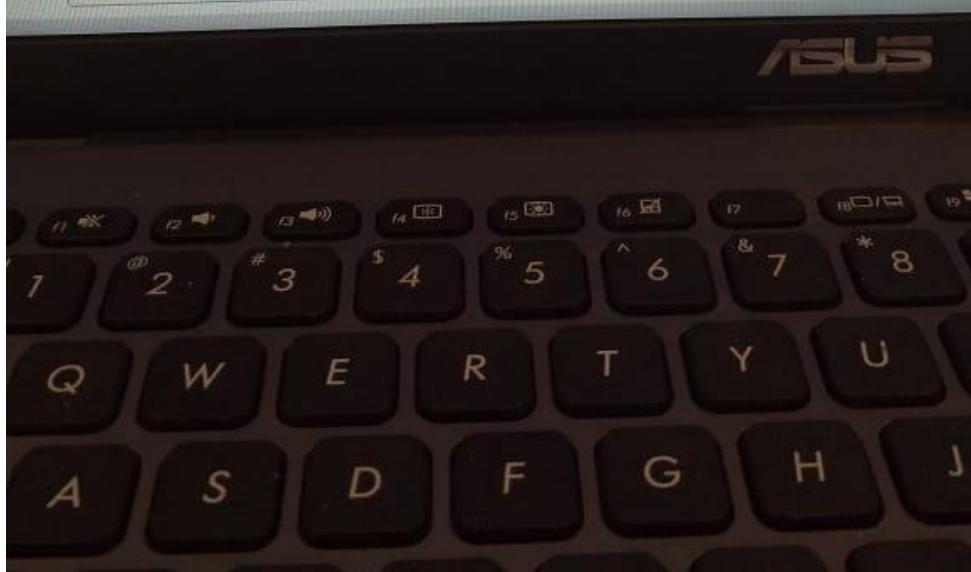
i A

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

```
class Main {  
    public static void main(String[] args) {  
        C1 c1 = new C1();  
        c1.method3(2, 3);  
    }  
}  
  
interface I1 {  
    void method1(int a, int b);  
    void method2(String a);  
}  
  
interface I2 {  
    void method3(int a, int b);  
    void method2(String a);  
}  
  
class C1 extends I1, I2 {  
    public void method1(int a, int b) {  
    }  
    public void method2(String a) {  
    }  
    public void method3(int a, int b) {  
    }  
}
```

Select one or more:

- C1 not implementing the method 2 twice
- Inheriting from multiple interfaces
- Both interfaces having the same method void method2(String a).
- Using the extends key word for inheritance of interfaces
- Creating a new C1 object



What are the errors in the following program?

```
class Test {  
    abstract int add(int a, int b);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "+" + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- The method public String add(String a, String b) is not applicable for the Test class reference
- Test class is not defined as an abstract class
- The abstract method in the Test class does not have an implementation
- Test2 class has not implemented the abstract method defined in Test class
- The method public String add(String a, String b) is not defined in the Test class



 NetExam
Sri Lanka Institute of Information Technology

fill the blank with the correct word

When you use final keyword for a method you cannot that method and when you use final keyword for a class you cannot create subclasses.

Answer:

[Next page](#)

Marked out of 1.00 [Flag question](#)

```
public static void main(String[] args) {
    try{
        try{
            int value = 10/0;
        }catch (ArithmetricException e) {
            System.out.println("ArithmetricException");
        }finally{
            System.out.println("Inner finally");
            int marks [] = new int[] {10};
            int value = marks[1];
        }
    }catch (NumberFormatException e) {
        System.out.println("NumberFormatException");
    }catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("ArrayIndexOutOfBoundsException");
    }finally {
        System.out.println("Outer finally");
    }
}
```

Select one:

- Line unreachable runtime error
- ArithmeticException
- Inner finally
- ArrayIndexOutOfBoundsException
- Outer finally
- ArithmeticException
- Inner finally
- Outer finally
- Compile Error
- ArithmeticException
- ArrayIndexOutOfBoundsException
- Outer finally

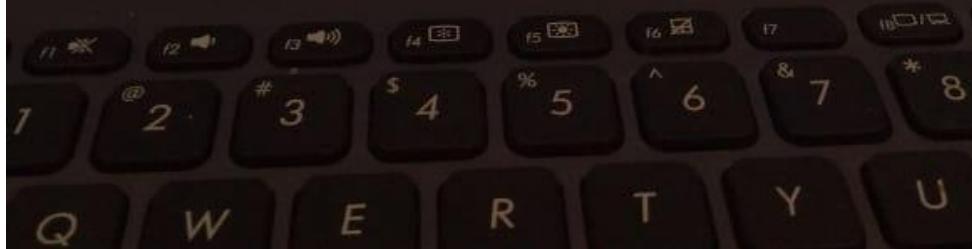
What are the errors in the following program?

```
abstract class Test {  
    abstract int add(int a, int b, int c);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a + "," + b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- Test2 class has not implemented the abstract method abstract int add(int a, int b, int c);
- We cannot create a Test2 object and assign it to a Test class reference.
- The main method is static and we cannot call a non-static method from a static method
- The method public String add(String a, String b) is not applicable for the Test class reference
- The method public int add(int a, int b) is not applicable for the Test class reference

ASUS





Question 19
Not yet answered
Marked out of 1.00
Flag question

Type the output of the below code.

```
StringBuilder sb = new StringBuilder("Sri Lanka Institute of Information Technology");
sb.reverse();
sb.substring(0, 5);
System.out.println(sb);
```

Answer:

[Next page](#)

Quiz navigation

[Finish attempt ...](#)

Time left 0:10:26

1

Mcq Questions

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20				

Structured Questions

21	22
----	----

Feedback

23

Question 13
Not yet answered
Marked out of
1.00
▼ Flag question

NetExam
Sri Lanka Institute of Information Technology

fill the blank with the correct word
When you use final keyword for a method you cannot
..... that method and when you use final keyword for a class
you cannot create subclasses.

Answer:

Next page

MCQ QUESTIONS

1	2	3	
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23

STRUCTURED QUESTIONS

FEEDBACK

Windows

The screenshot shows a web browser window with a dark blue header bar. In the top left corner of the header, there are three icons: a magnifying glass, a person icon, and a document icon. The main title 'NetExam' is displayed in a large, white, serif font. Below it, the subtitle 'Sri Lanka Institute of Information Technology' is shown in a smaller, white, sans-serif font.

Below the header, the content area has a light gray background. A question is posed in black text: "Refer the bellow code and choose the cases where line D will be executed ?".

```
//Line A
try {
    // Line B
} catch (SomeException se) {
    // Line C
} finally {
    // Line D
}
```

Following the code, a instruction "Select one or more:" is given, followed by a list of five options, each preceded by a small square checkbox:

- When Line B throws any type of exception
- When line A throws an exception
- When Line B throws an exception of sub class of SomeException
- When Line B throws an exception of type SomeException
- When Line B executed successfully

In the bottom right corner of the page, the letters "AS" are visible.

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- equal
- not equal
- Line 4 generates a compiler error.
- An exception is thrown on line 6.
- Line 1 generates a compiler error.



acer

F2	F3 Py0	F4 Z	F5 □■	F6 □■■	F7 □■■■	F8 □■■■■	F9	F10
#	\$	%	^	&	*	(
3	4	5	€	6	7	8		9

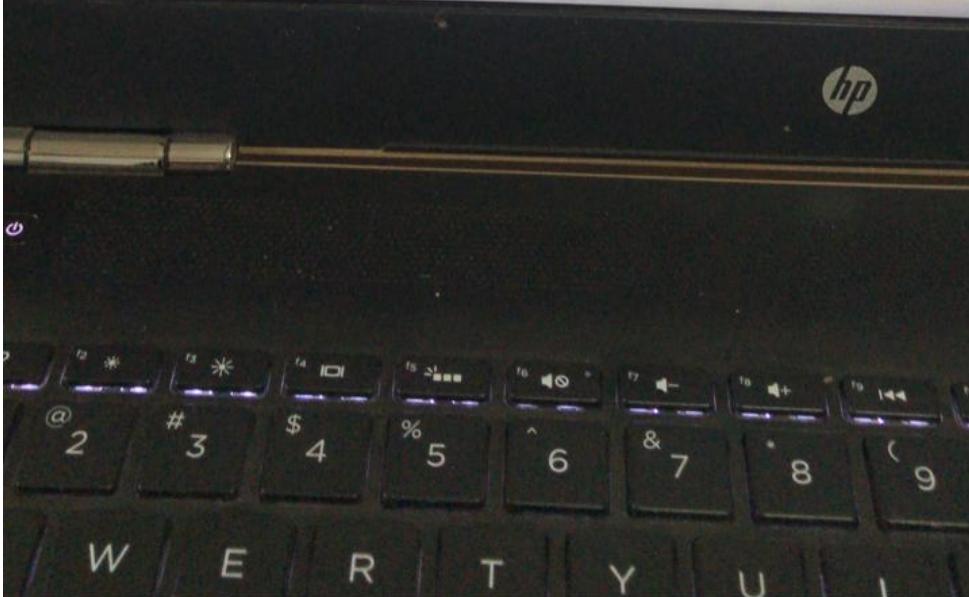
Sri Lanka Institute of Information Technology

Select the correct output of the following code snippet.

```
public static void main(String[] args) {  
    String result = new String("SLIIT Campus");  
    StringBuffer sb = new StringBuffer(result)  
        .toLowerCase().substring(0, 6));  
    System.out.println(result);  
    System.out.println(sb);  
}
```

Select one:

- SLIIT Campus
- SLIIT Campus
- slit c
- slit c
- SLIIT Campus
- slit
- slit campus
- slit
- SLIIT Campus



Question 18

Not yet answered

Marked out of
1.00

Flag question

Refer the bellow code and choose the cases where line D will be executed ?

```
//Line A
try {
    // Line B
} catch (SomeException se) {
    // Line C
} finally {
    // Line D
}
```

Select one or more:

- When line A throws an exception
- When Line B throws any type of exception
- When Line B executed successfully
- When Line B throws an exception of type SomeException
- When Line B throws an exception of sub class of SomeException

Sri Lanka Institute of Information Technology

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:

- S2 s2 = new S2(); is wrong as we cannot create an object from class S2
- 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
- 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

ASUS

Consider the below code:

```
public class ColorException extends Exception {}  
public class White {  
    public String printLine(String color) throws ColorExcept  
    {  
        if(color == null) { throw new ColorException (); }  
        return "Hello "+ color;  
    }  
}
```

You are adding below code in your application:

Line A : White white = new White()

Line B: System.out.println(white.printLine ("Ceramic"));

Select true answers from below:

Select one or more:

- Line B will compile if enclosed in a try block, where ColorException is caught.
- Line B can throw the unchecked exception ColorException
- Line A can throw the unchecked exception ColorException
- Line B will compile if the enclosing method throws a ColorException.
- Class White will not compile



DELL



NetExamination



Sri Lanka Institute of Information Technology

20

Answered
out of
question

Type the output of the below code.

```
StringBuilder sb = new StringBuilder("Sri Lanka Institute of Information Technology");  
sb.reverse();  
sb.substring(0, 5);  
System.out.println(sb);
```

Answer: alirs

Next Page

Question 12

yet answered

1 out of

Flag question

Given the following interface and class defined in a file named Traceable.java , what is the result of compiling this code?

```
1. public interface Traceable {  
2.     public static int MAX_TRACE;  
3.     public void trace();  
4. }  
5.  
6. class Picture implements Traceable {  
7.     public void trace() {  
8.         System.out.println("Tracing a picture");  
9.     }  
10. }
```

Select one:

- One bytecode file: Traceable.class
- Compiler error on line 3
- Compiler error on line 2
- Two bytecode files: Traceable.class and Picture.class
- Compiler error on line 6

NetExam

Sri Lanka Institute of Information Technology

Select the correct output of the following code snippet.

```
public static void main(String[] args) {  
    String result = new String("SLIIT Campus");  
    StringBuffer sb = new StringBuffer(result.toUpperCase()  
        .toLowerCase().substring(0, 6));  
    System.out.println(result);  
    System.out.println(sb);  
}
```

Select one:

- SLIIT Campus
- slit
- SLIIT Campus
- SLIIT Campus
- slit campus
- slit
- sliit c
- silit c

Question 10
Not yet answered
Marked out of
1.00
 Flag question

What are the errors in the following program?

```
class Test {  
    abstract int add(int a, int b);  
}  
  
class Test2 extends Test {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public String add(String a, String b) {  
        return a +" "+ b;  
    }  
}  
  
class Main{  
    public static void main(String[] args) {  
        Test t2 = new Test2();  
        System.out.println( t2.add(2,3));  
        System.out.println( t2.add("abc","def"));  
    }  
}
```

Select one or more:

- The method public String add(String a, String b) is not defined in the Test class
- The method public String add(String a, String b) is not applicable for the Test class reference
- Test2 class has not implemented the abstract method defined in Test class
- The abstract method in the Test class does not have an implementation
- Test class is not defined as an abstract class

idle

NetExam
Sri Lanka Institute of Information Technology

18
answered
out of
question

Refer the bellow code and choose the cases where line D will be executed?

```
//Line A
try {
    // Line B
} catch (SomeException se) {
    // Line C
} finally {
    // Line D
}
```

Select one or more:

- When Line B throws an exception of type SomeException
- When line A throws an exception
- When Line B throws an exception of sub class of SomeException
- When Line B executed successfully
- When Line B throws any type of exception

DELL

F1 F2 F3 F4

Sri Lanka Institute of Information Technology

in 18
answered
d out of
g question

Select the correct Exception output of the following program.

```
public static void main(String[] args) {
    try{
        try{
            int value = 10/0;
        }catch (ArithmetricException e) {
            System.out.println("ArithmetricException");
        }finally{
            System.out.println("Inner finally");
            int marks [] = new int[] {10};
            int value = marks[1];
        }
    }catch (NumberFormatException e) {
        System.out.println("NumberFormatException");
    }catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("ArrayIndexOutOfBoundsException");
    }finally {
        System.out.println("Outer finally");
    }
}
```

Select one:

- Line unreachable runtime error
- ArithmetricException
ArrayIndexOutOfBoundsException
Outer finally
- ArithmetricException
Inner finally
Outer finally
- ArithmetricException
Inner finally
ArrayIndexOutOfBoundsException
Outer finally
- Compile Error

NetExam

Sri Lanka Institute of Information Technology

What is the result of the following code?

```
String city = null;  
if(city.equals("Boston")) {  
    System.out.print("true");  
}else {  
    System.out.print("false");  
}finally {  
    System.out.print("finally");  
}
```

Select one or more:

- The code does not compile.
- false
- finally
- true
- finally , followed by the stack trace from a NullPointerException

ASUS

```
        }
    finally {
        System.out.print("finally ");
    }
}
public static void main(String[] args) {
try {
    test();
}
catch (Exception ex) {
    System.out.print("exception ");
}
}
Select one:
 a. Compilation fails
 b. Null
 c. Null finally
 d. Finally exception
 e. Finally
```

Consider the below code:

```
public class ColorException extends Exception { }
public class White {
    public String printLine(String color) throws ColorException
    {
        if(color == null) { throw new ColorException(); }
        return "Hello "+ color;
    }
}
```

You are adding below code in your application:

Line A : White white = new White()

Line B: System.out.println(white.printLine ("Ceramic"));

Select true answers from below:

Select one or more:

- Line A can throw the unchecked exception ColorException
- Line B will compile if the enclosing method throws a ColorException.
- Line B will compile if enclosed in a try block, where ColorException is caught.
- Line B can throw the unchecked exception ColorException
- Class White will not compile

ASUS VivoBook



answered
1 out of
8 question

Consider the below code:

```
public class ColorException extends Exception { }
public class White {
    public String printLine(String color) throws ColorException
    {
        if(color == null) { throw new ColorException (); }
        return "Hello "+ color;
    }
}
```

You are adding below code in your application:

Line A : White white = new White();

Line B: System.out.println(white.printLine("Ceramic"));

Select true answers from below:

Select one or more:

- Class White will not compile.
- Line A can throw the unchecked exception ColorException.
- Line B will compile if the enclosing method throws a ColorException.
- Line B will compile if enclosed in a try block, where ColorException is caught.
- Line B can throw the unchecked exception ColorException

≡ Quiz

Finish att

Time left 0

1

MCQ QUE

1 2

8 9

15 16

STRUCTUR

21 22

FEEDBACK

23

What is the result of the following code?

```
1 byte twelve = -12;
2 Byte b1 = new Byte(twelve);
3 Byte b2 = new Byte(twelve);
4 if(b1.byteValue() == b2) {
5     System.out.println("equal");
6 }
7 else {
8     System.out.println("not equal");
9 }
10
```

Select one:

- An exception is thrown on line 6.
- Line 4 generates a compiler error.
- Line 1 generates a compiler error.
- not equal
- equal

DELL





Question 19

Not yet answered

Marks out of
0.00

Flag question

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);
```

insert append 4 1

4 1 insert append

insert 4 append 1