

Ques	Ans	Option A	Option B	Option C	Option D
class Shape{ public Shape(){ } } class Rectangle extends Shape{ } class MainClass { public static void main(String [args]) { Shape shape = new Shape(); //line 1 rectangle rect = (Rectangle)shape; //line 2 }	This code will compile and run successfully. This will generate ClassCastException as shape can not be typecasted to Rectangle.	It will work fine if line 1 will be replaced by below given code: Shape shape = new Rectangle();	It will work fine if line 1 will be replaced by below given code: Shape shape = new Rectangle();	Both 2 and 3 are correct.	Both 2 and 3 are correct.
What is the true statement about above code?					
public class Employee implements Comparable<Employee>{ int empId; String empName; }	Employee objects will be sorted as per id	It will generate ClassCastException	It will sort Employee objects as per id if class Employee will implement Comparable and override compareTo method	Both 2 and 3 are correct.	Both 2 and 3 are correct.

Cut Copy Paste

Calibri (Body) 11 A A Alignment Number

Format as Table Cell Styles

Share Search Options/ Answer 1 Options/ Answer 2 Options/ Answer 3 Options/ Answer 4

E12 G JDBC

H

J

K

Q

Text

W

```
class Shape{  
    public Shape(){  
    }  
}  
class Rectangle extends Shape{  
}  
class MainClass  
{  
    public static void main(String []args)  
    {  
        Shape shape = new Shape(); //line 1  
        Rectangle rect = (Rectangle)shape; //line 2  
    }  
}
```

What is the true statement about above code?

```
public class Employee implements  
Comparator<Employee>{  
  
    int empId;  
    String empName;  
    .....
```

This code will compile and run successfully

This will generate ClassCastException as shape can not be typecasted to Rectangle

It will work fine if line 1 will be replaced by below given code
Shape shape = new Rectangle();

Both 2 and 3 are

Employee objects will be sorted as per id

It will generate ClassCastException

It will sort Employee objects as per id if class Employee will implement Comparable and override compareTo method

Both 2 and 3 are

Sheet1

Sheet2 Sheet3 +

Module2_40Questions

Search Sheet

Home

Insert

Page Layout

Formulas

Data

Review

View

Calibri (Body)

11

A A

Alignment

Number

Conditional Formatting

Format as Table

Cells

Editing

Share

Q

Search

Module2_40Questions

Search Sheet

Share

2 X ✓ fx JDBC

G

H

I

J

K

Q text

Employee objects will be sorted as per id

Option/ Answer 1

It will generate ClassCastException

Option/ Answer 2

Both 2 and 3 are correct

Option/ Answer 3

None of the above

```
public class Employee implements Comparator<Employee>{  
    int empId;  
    String empName;  
    //getters & setters  
    public Employee(int empId, String empName) {  
        super();  
        this.empId = empId;  
        this.empName = empName;  
    }  
    @Override  
    public int compare(Employee emp1,  
                      Employee emp2){  
        return emp1.getEmpId()-emp2.getEmpId();  
    }  
  
    public class MainClass{  
        public static void main(String[] args){  
            System.out.println("Hello World");  
        }  
    }  
}
```

1

Sheet2

Sheet3

115%

<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Employee objects will be sorted as per id

Option/ Answer 1	Option/ Answer 2	Option/ Answer 3
10 tea 20 coffee	It will generate NumberFormatException	tea coffee tea

```

import java.util.*;
public class ScannerDemo {

public static void main(String[] args) {

String input = "10 tea 20 coffee 30 tea biscuits";
Scanner sc = new Scanner(input).useDelimiter("\s");
System.out.println(sc.next());
System.out.println(sc.next());
System.out.println(sc.next());
System.out.println(sc.next());
sc.close();
}
}

```

What will be the output of above code?

Which of the given method is used to remove

Sheet1
Sheet2
Sheet3

115%

1	Q) ArrayList System.out.println(sc.nextInt()); System.out.println(sc.nextInt()); sc.close(); }						
6	 What will be the output of above code?						
7	 Which of the given method is used to remove element from the list? remove(position) delete(position)						
8	 Which of the given class uses hashCode value of an object to determine, how the object should be stored in the collection? class Employee { int empId;						

```
class Employee  
{  
    int empid;  
    String empname;  
    static int count;  
    public Employee(int id, String name)  
    {  
        empid = id;  
        empname = name;  
        count++;  
    }  
}
```

```
class MainClass  
{  
    public static void main(String args)  
    {  
        try{  
            FileOutputStream out = new  
FileOutputStream("d:\\Employee.txt");  
            ObjectOutputStream oos = new  
ObjectOutputStream(out);  
            Employee e1 = new Employee(1, "Rahul");  
            Employee e2 = new Employee(2, "Karan");  
            Employee e3 = new Employee(3, "Aman");  
            Employee e4 = new Employee(4, "Vishal");  
            Employee e5 = new Employee(5, "Rohit");  
            oos.writeObject(e1);  
            oos.writeObject(e2);  
            oos.writeObject(e3);  
            oos.writeObject(e4);  
            oos.writeObject(e5);  
            oos.close();  
        }  
        catch(Exception e){  
            System.out.println("Exception "+e);  
        }  
    }  
}
```

It will serialize employee object i.e. empid, empname and count will be stored in the file.

It will serialize only empid and empname.

It will generate error in ObjectOutputStream.

```
String empName;  
static int count;  
public Employee(int id, String name) {  
    empld = id;  
    empName = name;  
    count++;  
}
```

```
Class MainClass {  
    public static void main(String[] args) {  
        try {  
            FileOutputStream out = new FileOutputStream("d:\\serl.txt");  
            ObjectOutputStream bout = new ObjectOutputStream(out);  
            Employee e1 = new Employee(111,"RAM");  
            bout.writeObject(e1);  
            bout.flush();  
            System.out.println("done");  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Which of the given statement is true about above code?"

669 characters in 25 lines



1	Q text	G	H	I	J
	public class StreamAPIDemo {		Option Answer 1	Option Answer 2	Option Answer 3
10	public static void main(String[] args) {				
	String []arr = new String[]{“Capgemini”, “Global”, “Solution”, “Mumbai”};				
	Stream<String>>str = Arrays.stream(arr);				
	String msg = str.filter(str->				
	>str.contains(“o”)).reduce((str1,str2)->str1.concat(str2)).get();				
	System.out.println(msg);				
	}				
	What will be the output of above code?				
	interface CalculateLength				
	{				
	int length(String str);				
	}				
	public class LambdaDemo				
	{				

Sheet1	Sheet2	Sheet3	+	
GlobalSolution	GlobalSolution	GlobalSolution		Capgemini/Global/Soluti mbu

1 Q Text

```
interface CalculateLength
{
    int length(String str);
}

public class LambdaDemo
{
    public static void main(String []args)
    {
        //line1
        System.out.println(ref.length() + "capgemini");
    }
}Which of the given option is correct to insert at line1 for the
successful execution of the code?
```

Option/ Answer 1

Option/ Answer 2

Option/ Answer 3

```
CalculateLength ref =
(str1->str1.length());
```

```
int ref = (str1-
>str1.length());
```

```
CalculateLength ref = str1
```

```
//line1
System.out.println(ref.length() + "capgemini");
```

11

```
public class JDBCdemo{
    public static void main(String[] args) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con =
                DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:X");
        }
    }
}
```

Sheet1

Sheet2

Sheet3

+



115%

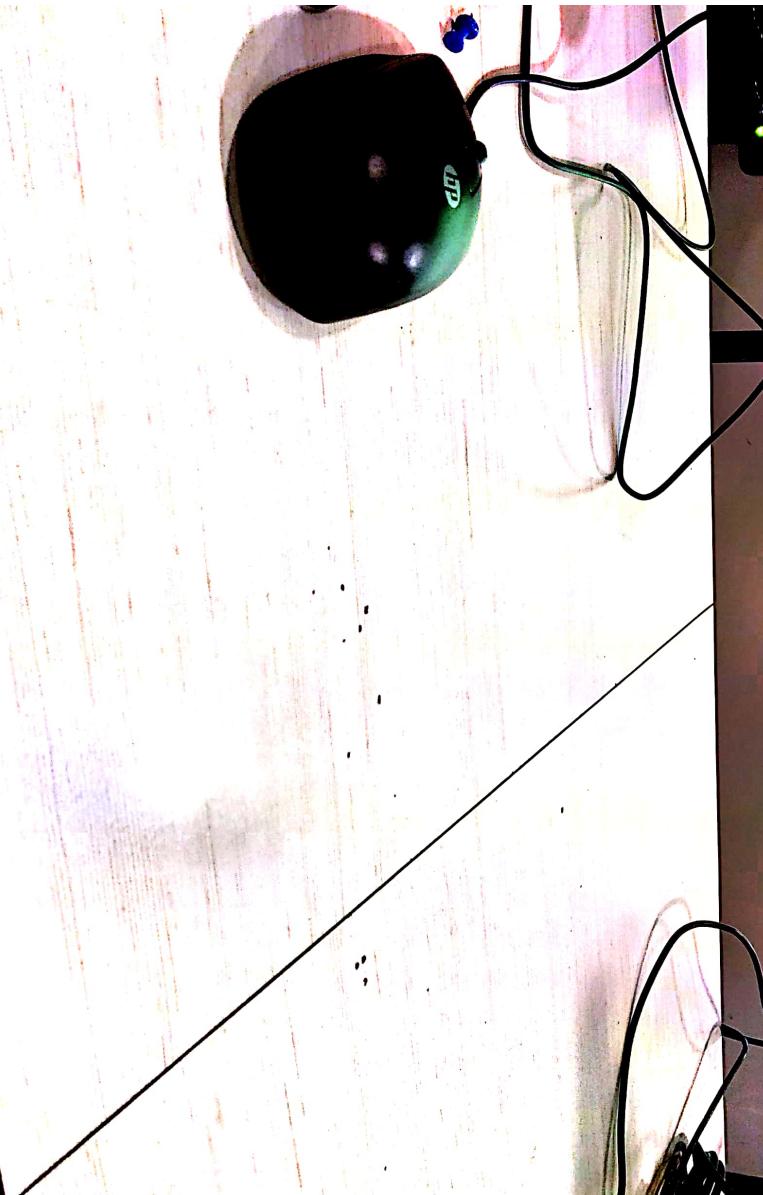
```

public class JDBCdemo {
    public static void main(String[] args) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con =
                DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:X
E","system","sys");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM Emp");
            rs.next();
            while(rs.next())
            {
                System.out.println(rs.getInt(1));
                System.out.println(rs.getString(2));
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            --
```

It will print records from
second row onwards

No output

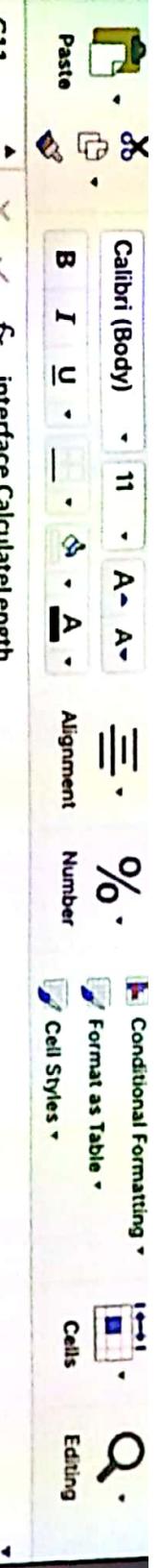
It will throw SQLException
rs.next() is called twice



```
1  ▶ D Test
public static void main(String[] args) {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system", "sys");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM Emp");
        rs.next();
        while(rs.next()){
            System.out.println(rs.getInt(1));
            System.out.println(rs.getString(2));
        }
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Option Answer 1	Option Answer 2	Option
It will print records from second row onwards.	No output	It will print all the rows.





Paste

X

B I U — A A Alignment

Number

Format as Table

Cell Styles

Cells

Editing

Trebuchet MS 10 A A %

115 No output

G H

Option/Answer 1 Option/Answer 2 Option/Answer 3

Q text

System.out.println(rs.getString(1));
System.out.println(rs.getString(2));
}
}
catch (ClassNotFoundException e) {
 e.printStackTrace();
}
} catch (SQLException e) {
 e.printStackTrace();
}

Which of the given statement interface supports input output parameters?

Arrange the below given steps to call procedure/function

1)Create callable statement object

2)Call setXXX() method to set IN parameters

3)Call execute() to invoke the procedure/function

4)Call getXXX() method to retrieve results from OUT parameters/function return value

5)Call registerOutParameter() method to register OUT parameters/function return value

public class ExceptionDemo {

Sheet1 Sheet2 Sheet3 +

- + 115%

Statement

PreparedStatement

Callable

1,2,5,3,4

1,2,3,4,5

2,1,3,4

Q text

```
System.out.println(rs.getString(1));
System.out.println(rs.getString(2));
}
}
}
catch (ClassNotFoundException e) {
    e.printStackTrace();
}
catch (SQLException e) {
    e.printStackTrace();
}
```

Option/ Answer 1

Option/ Answer 2

Option

Which of the given statement interface supports input output parameters?

Statement

PreparedStatement

Callable

Arrange the below given steps to call procedure?

- 1)Create callable statement object
- 2)Call setXXX() method to set IN parameters
- 3)Call execute() to invoke the procedure/function
- 4)Call getXXX() method to retrieve results from OUT parameters/function return value
- 5)Call registerOutParameter() method to register OUT parameters/function return value

Sheet1 Sheet2 Sheet3 +

115%

1 Q Text

Option/ Answer 1

Option/ Answer 2

Option/ Answer 3

...

```
public class ExceptionDemo {  
    public static void main(String[] args) {  
        int i = 0;  
        try {  
            i = Integer.parseInt("abc");  
        } catch(Exception ex) {  
            System.out.println("Exception");  
        } finally {  
            System.out.println("i = "+i);  
        }  
    }  
}
```

What will be the output of above code?

Exception
i=0

i = abc

no output

They represent

18

Sheet1

Sheet2

Sheet3

+

– + 115%

ANSWER

Paste

Calibri (Body)

11

A A

Alignment

Number

Cell Styles

Conditional Formatting

Format as Table

Cells

Editing

H17

1,2,5,3,4

G

H

I

Q text

Option/ Answer 1

Option/ Answer 2

Option

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

try{
String str = null;
int len = str.length();

} catch(Exception ex)

{
System.out.println("In catch ");
throw ex;



What will be the output of above code?

20
Which of the given keyword describes the exception, which can be raised by a method?

21
class MyThread implements Runnable{
public void run()

finally

throw

throws

In Catch followed by
Exception stack trace

In Catch

No Ex

115%

Sheet1

Sheet2

Sheet3

+



Q text

Option/ Answer 1

Option/ Answer 2

Option/ Answer 3

```
class MyThread implements Runnable{
```

```
public void run(){
```

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

```
}
```

```
public class ThreadDemo
```

```
{
```

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

```
Thread thread = new Thread(new MyThread());
```

```
thread.start(); //line 1
```

```
thread.start(); //line 2
```

} which of the given is a true statement about above code?

23

```
public class Calculator {
```

```
public int add(int num1,int num2)
```

```
{
```

```
return num1+num2;}
```

```
public int div(int num,int den)
```

}
This code will generate output Hi followed by java.lang.IllegalThreadStateException due to line 2

This code will generate output +

This code will generate output +

This code will generate output +

Hi

Hi

Hi

Calculator ref;

@Before

public void init()

{
ref = new Calculator();}

@Ignore

@Test

public void testAdd()

{
assertEquals(0,ref.add(3, 4));

}

@Test(expected=ArithmaticException.class)

public void testDiv()

{
ref.div(5, 2);

})

What will the output after the execution of CalculatorTest?

java.lang.AssertionError
for both the test

methods

java.lang.AssertionError
for testDiv

jav

test

class TestEmployeeDao

class TestEmployeeDao

Sheet1 Sheet2 Sheet3 +

115%

1 Q base

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Option / Answer 1

Option / Answer 2

Option / Answer 3

Option / Answer 4

```
class EmployeeDao
{
    public Employee insertEmployee(Employee emp)
    {
        //jdbc insert code
    }
}
```

Which of the given Test class is correct to test the insertEmployee() method and to initialize Dao object only once before all the test case execution?

25

EmployeeName

Sheet1

Sheet2

Sheet3

+

115%

```
class TestEmployeeDao
{
    @BeforeClass
    public static void init()
    {
        EmployeeDao dao = new EmployeeDao();
        dao.insertEmployee(emp);
    }

    @Test
    public void testInsertEmployee()
    {
        Employee emp = new Employee();
        emp.setName("John");
        emp.setAge(25);
        emp.setSalary(10000);
        EmployeeDao dao = new EmployeeDao();
        dao.insertEmployee(emp);
        assertEquals("John", emp.getName());
        assertEquals(25, emp.getAge());
        assertEquals(10000, emp.getSalary());
    }
}
```


27

public class LambdaDemo {

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

```
Supplier<String> supplier = ()->"Welcome to Lambda Expression";
```

```
Predicate<String> pred = (str1)->str1.length()>10;
```

```
System.out.println(pred.test(supplier.get()));}
```

log4j.appenders.FILE.layout.conversionPattern=%m%n
layout=org.apache.log4j.SimpleLayout

log4j.appenders.loginfo.layout.conversionPattern=%m%n
layout=org.apache.log4j.SimpleLayout

log4j.appenders.loginfo.layout.conversionPattern=%m%n
layout=org.apache.log4j.SimpleLayout

28

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

```
Integer[] arr = new Integer[]{29,4,45,6,11,14,8,13};
```

```
Stream<Integer> sArr = Arrays.stream(arr);
```

```
sArr.limit(6).filter((num)->num>4).forEach((num)->System.out.print(num));
```

29,45,6,11,14

29,4,45,6,11,14

29,4,45,

TRUE

FALSE

compila

29

}**}**

Sheet1 Sheet2 Sheet3 +

14 115%

```

public class FileDemo {
    public static void main(String args) {
        BufferedReader br = null;
        String strLine = "";
        try {
            br = new BufferedReader( new FileReader("d:\\reader.txt") );
            while( (strLine = br.readLine()) != null){
                System.out.println(strLine);
            }
        } catch (FileNotFoundException e) {
            System.err.println("Unable to find the file: fileName");
        } catch (IOException e) {
            System.err.println("Unable to read the file: fileName");
        }
    }
}

```

30

Which of the given statement is true about above code?

It will read the contents
of reader.txt line by line

It will read the contents
of reader.txt character
by character

It will re...
ader.t

Calibri (Body) 11 A A Alignment Number % Conditional Formatting Format as Table Cells Editing

Paste Copy Cut B I U _ A Alignment Number Cell Styles

H28 ▾ X ✓ fx TRUE

G

H

Q text

```
//getters & setters  
class Manager extends Employee
```

```
//getters & setters  
public class MainClass
```

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

```
{  
Manager obj = new Manager();  
System.out.println(obj instanceof Manager);  
System.out.println(obj instanceof Employee);  
System.out.println(obj instanceof Object);  
}
```

31)What will be the output of MainClass?

true
true
true
false
false
false
true
true
false

Option/ Answer 1 Option/ Answer 2 Option/ Answer 3

32 public class LocalDateDemo {

Which of the given String class method converts data to string?

concat()
substring()
value

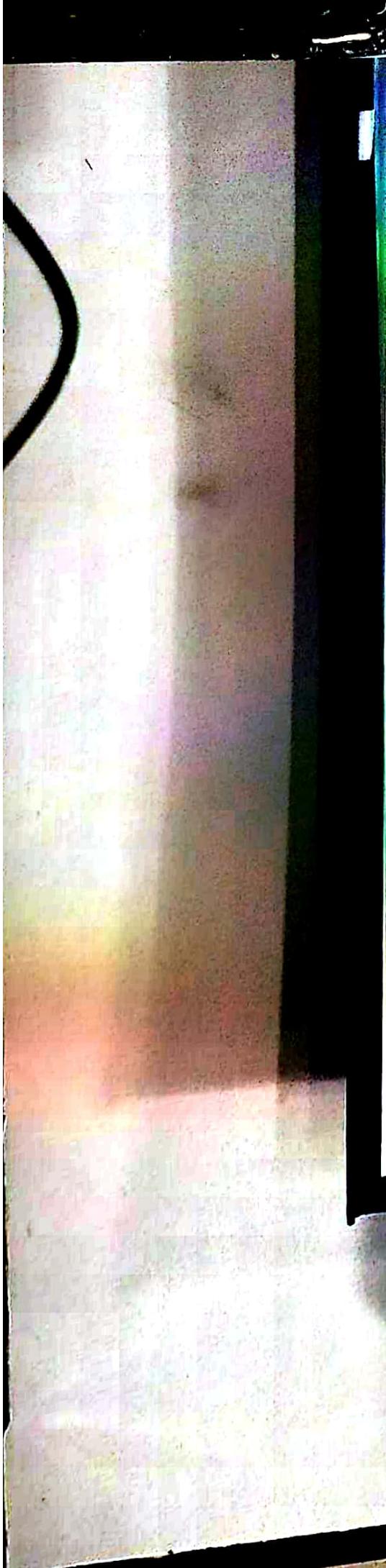
Sheet1 Sheet2 Sheet3 +

115%

1	Q text	G	H
34	What will be the output of above code?	ArrayList<Integer>list = new ArrayList<Integer>(); list.add(1); list.add(2); list.add(3); System.out.println(list);	111
35	What will be the output of above code?	Hashtable<Integer, String>map = new Hashtable<Integer, String>(); map.put(1, "Global"); map.put(2, "Capgemini"); map.put(3, "Solution"); Set<Entry<Integer, String>> set = map.entrySet(); Object[] arr = set.toArray(); for (Object val : arr) { System.out.println(val); }	101-Gaggevni 102-Globa 103-Solutions Gaggevni Global Solutions 101 102 103

J33 It is will generate DateTimeParseException as format and input are not matching

G	H	I
1	Option/ Answer 1	Option/ Answer 2
35 Which of the given option is correct to convert map keys into list?	<pre>List<Integer>list = new ArrayList<Integer>(map.keySet());</pre>	<pre>List<Integer>list = new ArrayList<Integer>(map.entrySet());</pre>
36 Implementation of Comparable interface is mandatory to user defined class when the same class objects are added to—	TreeSet	HashSet
37 ? super T stands for?	any reference types which are subclass of T are allowed.	any reference types which are subclass of T are allowed.
38 Which of the given code is correct to accept and print the string and also to compare two numbers using built in functional interface?	<pre>Consumer<String> consumer = (String str) -> System.out.println(str); consumer.accept("Hello World"); BiPredicate<Integer, Integer> predicate =</pre>	<pre>Supplier<String> supplier = (String str) -> System.out.println(str); supplier.supply("Hello World"); Predicate<Integer, Integer> predicate =</pre>



37

int the string and also to compare
I are allowed.

```
Consumer<String> consumer = new Consumer<String>() {
    @Override
    public void accept(String str) {
        System.out.println(str);
    }
};

Supplier<String> supplier = new Supplier<String>() {
    @Override
    public String get() {
        return "Hello World";
    }
};

BiPredicate<Integer, Integer> predicate = (num1, num2) -> (num1 > num2);
BiPredicate<Integer, Integer> predicate = (num1, num2) -> (num1 < num2);

System.out.println(predicate.test(45, 2));
System.out.println(predicate.test(45, 2));
```

38

```
me = "Hello me";
(str)->System.out.println(str);
```

```
Supplier<String> supplier = new Supplier<String>() {
    @Override
    public String get() {
        return "Hello World";
    }
};

BiPredicate<Integer, Integer> predicate = (num1, num2) -> (num1 > num2);
BiPredicate<Integer, Integer> predicate = (num1, num2) -> (num1 < num2);

System.out.println(predicate.test(45, 2));
System.out.println(predicate.test(45, 2));
```

Sheet2 Sheet3 +

115%

Hello
Hi
Welcome

Hello
Hi
Welcome

Hi
Welcome

Paste

Cut

Trebuchet MS 10 A A Alignment Number

Format as Table Cells

Conditional Formatting Cell Styles Editing

N37 □ X ✓ Fx 1

G

H

I

Q text

Option/ Answer 1

Option/ Answer 2

Option

public class StreamDemo {

```
public static void main(String[] args) {  
    Stream<String>str = Stream.of("Hello", "Hi", "Welcome");  
    str.filter((ref)->ref.contains("I")).distinct().forEach((str)->System.out.println(str));  
}
```

What is the output of above code?



Hello
Welcome

Hello
Hi
Welcome

Hello
Hi

39 public class StreamDemo {

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

```
BiFunction<String, String>fun = (str1,str2)->str1+str2;  
String result=null;  
result = fun.apply("10", "20");  
System.out.println(result);  
}
```

40 What will be the output of above code?

1020

30

compil:

Sheet1

Sheet2

Sheet3

+

115%

