IRC_JAVA_CA1_SET1_updated

Test Summary

- No. of Sections: 2
- No. of Questions: 25
- Total Duration: 90 min

Section 1 - MCQ

Section Summary

- No. of Questions: 20
- Duration: 30 min

Additional Instructions:

None

Q1. What are different types of Inheritance supported by Java?

- Multiple Inheritance
- Multi-level inheritance Only
- Hybrid Inheritance
- All of the above

Q2. Consider the Counter class below and answer the question that follows.

Using the class above and the variable declared below, what is the value of num.toString()?

```
Counter num = new Counter();
```

```
1  public class Counter
2  {
3     public int count = 0;
4     public int getCount()
5     {
6        return count;
7     }
8     public void increment()
9     {
10       count++;
11    }
12 }
13
```

- a string with count's value
- a string with num's type and hashcode
- a string with count's type and hashcode
- nothing since toString is not defined for Counter

Q3. Consider the following inheritance hierarchy diagram:

Which of the following statements is correct?

Auto is a superclass of LandVehicle, and LandVehicle is a superclass of Vehicle.

Auto is a superclass of LandVehicle, and LandVehicle is a subclass of Vehicle.

Auto is a subclass of LandVehicle, and LandVehicle is a superclass of Vehicle.

Auto is a subclass of LandVehicle, and LandVehicle is a subclass of Vehicle.

Q4. Which statement about a no-argument constructor is true?

The Java compiler will always insert a default no-argument constructor if you do not define a no-argument constructor in your class.

In order for a class to call super() in one of its constructors, its parent class must explicitly implement a no-argument constructor.

If a class extends another class that has only one constructor that takes a value, then the child class must explicitly declare at least one constructor.

A class may contain more than one no-argument constructor.

Q5. Which of the following is a method having same name as that of it’s class?

finalize

delete

class

constructor

Q6. What would be behaviour if constructor has a return type?

Compilation error

Runtime error

Compilation and runs successfully

Only String return type is allowed

Q7. What is the output of the following code?

```
1  public class Null
2  {
3     public static boolean greet()
4   {
5         System.out.print("Program execute = ");
6         return true;
7   }
8     public static void main(String[] args)
```

```
 9  {
10      System.out.println(Null.greet());
11  }
12  }
13
```

- Program execute = false
- Program execute = true
- Program execute = 1
- Program execute = 0

Q8. If there are 3 classes. Class C is derived from class B and B is derived from A, Which class destructor will be called at last if object of C is destroyed.

- A
- B
- C
- All together

Q9. If there are 5 classes, E is derived from D, D from C, C from B and B from A. Which class constructor will be called first if the object of E or D is created?

- A
- B
- C
- A and B

Q10. Predict the behavior of the following code.

```
1  interface IShape {
2     void f1();
3     void f2();
4     void f3();
5  }
6  class Circle implements IShape {
7     public void f1() {
8     }
9  }
10
```

- Compile time error
- Run time error
- The code is correct

Exception

Q11. Which is correct option about java interface?

Interface is used to achieve multiple inheritance in java.

Object of an interface cannot be created.

An interface can extend another interface.

All of the above.

Q12. What is output of the following program?

```
1  class student
2  {
3     public : int marks;
4     void disp()
5     {
6        cout<<”Its base class”
7     };
8     class topper:public student
9     {
10       public :
11       void disp()
12       {
13          cout<<”Its derived class”;
14       }
15    }
16    void main() { student s; topper t;
17    s.disp();
18    t.disp();
19 }
```

Its base classIts derived class

Its base class Its derived class

Its derived classIts base class

Its derived class Its base class

Q13. Which among the following best describes polymorphism?

It is the ability for a message/data to be processed in more than one form

It is the ability for a message/data to be processed in only 1 form

It is the ability for many messages/data to be processed in one way

It is the ability for undefined message/data to be processed in at least one way

Q14. How to format date from one form to another?

- SimpleDateFormat
- DateFormat
- SimpleFormat
- DateConverter

Q15. Which statement about methods in an interface is true?

- All methods in an interface are automatically private.
- All methods in an interface are automatically public.
- All methods in an interface are automatically static.
- All methods in an interface must be explicitly declared as private or public.

Q16. What are the class variables in the following program?

```
1  import java.io.*;
2  import java.util.*;
3  public class IdentifyMyParts {
4     public static int x = 7;
5     public int y = 3;
6  }
7  public class Main {
8     public static void main(String [] args) {
9        IdentifyMyParts a = new IdentifyMyParts();
10       IdentifyMyParts b = new IdentifyMyParts();
11       a.y = 5;
12       b.y = 6;
13       a.x = 1;
14       b.x = 2;
15       System.out.println("a.y = " + a.y);
16       System.out.println("b.y = " + b.y);
17       System.out.println("a.x = " + a.x);
18       System.out.println("b.x = " + b.x);
19       System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
20 }
21
```

- x y
- x
- a
- a b

Q17. What will be the output of the following code?

```
1  import java.io.*;
2  import java.util.*;
3  public class IdentifyMyParts {
4     public static int x = 7;
5     public int y = 3;
6  }
7  public class Main {
8     public static void main(String [] args) {
9        IdentifyMyParts a = new IdentifyMyParts();
10       IdentifyMyParts b = new IdentifyMyParts();
11       a.y = 5;
```

```
11          a.y = 5;
12          b.y = 6;
13          a.x = 1;
14          b.x = 2;
15          System.out.println("a.y = " + a.y);
16          System.out.println("b.y = " + b.y);
17          System.out.println("a.x = " + a.x);
18          System.out.println("b.x = " + b.x);
19          System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
20      }
21      }
```

- a.y = 5
b.y = 6
a.x = 2
b.x = 2
IdentifyMyParts.x = 2
- Run time error
- Compile time error
- a.y = 6
b.y = 5
a.x = 2
b.x = 2
IdentifyMyParts.x = 2

Q18. What are the instance variables in the program?

```
1   import java.io.*;
2   import java.util.*;
3   public class IdentifyMyParts {
4       public static int x = 7;
5       public int y = 3;
6   }
7   public class Main {
8       public static void main(String [] args) {
9           IdentifyMyParts a = new IdentifyMyParts();
10          IdentifyMyParts b = new IdentifyMyParts();
11          a.y = 5;
12          b.y = 6;
13          a.x = 1;
14          b.x = 2;
15          System.out.println("a.y = " + a.y);
16          System.out.println("b.y = " + b.y);
17          System.out.println("a.x = " + a.x);
18          System.out.println("b.x = " + b.x);
19          System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
20      }
21      }
```

- x
- x y
- a
- a b

Q19. A class that cannot be instantiated is called a/an ____.

- Abstract class.

Anonymous class.

Concrete class.

Non-inheritable class.

Q20. What is the output of this program?

```
1  class Mammal
2  {
3     void eat(Mammal m)
4     {
5        System.out.println("Mammal eats food");
6     }
7  }
8  class Cattle extends Mammal
9  {
10    void eat(Cattle c)
11    {
12       System.out.println("Cattle eats hay");
13    }
14 }
15 class Horse extends Cattle
16 {
17    void eat(Horse h)
18    {
19       System.out.println("Horse eats hay");
20    }
21 }
22 public class Test
23 {
24    public static void main(String[] args)
25    {
26       Mammal h = new Horse();
27       Cattle c = new Horse();
28       Horse s = new Horse();
29       c.eat(c);
30    }
31 }
32
```

prints "Mammal eats food"

prints "Cattle eats hay"

prints "Horse eats hay"

Class cast Exception at runtime

Section 2 - CODING

Section Summary

- No. of Questions: 5
- Duration: 60 min

Additional Instructions:

None

Q1. Write a program to print the area and perimeter of a rectangle by creating a class named "Rectangle".

Input Format

The input consists of the length and breadth of a rectangle.

Output Format

The output prints the area and perimeter of the rectangle.

Sample Input

```
20 50
```

Sample Output

```
1000
140
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Write a java program to find the sum of two numbers without using the **+** operator.

Sample Input

```
618
229
```

Sample Output

```
847
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Write a function to find Fibonacci numbers one by one within a limit **N.** Let 2 and 3 are the initial numbers in the sequence. Print the number. Pass it to another function that checks if the number is a power of 2. If so, print the number. Create a class that has these two functions and the main() function.

Input Format

The input consists of the value n.

Output Format

The output prints the Fibonacci numbers and whether the number is the power of two or not.
Refer to the sample output for formatting specifications.

Sample Input

```
5
```

Sample Output

```
2 is a fibonacci number
2 is a power of two
3 is a fibonacci number
5 is a fibonacci number
```

Sample Input

```
15
```

Sample Output

```
2 is a fibonacci number
2 is a power of two
3 is a fibonacci number
5 is a fibonacci number
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. **Overriding-simple**

Overriding is another concept that every application developer should know. Overriding is a runtime polymorphism. The inherited class has the overridden method which has the same name as the method in the parent class. The argument number, types or return types should not differ in any case. The method is invoked with the object of the specific class (but with the reference of the parent class).
Now let's try out a simple overriding concept in our application. For this, we can take our original example of Class Event, and its child classes Exhibition and StageEvent.
Create a parent class **Event** with following protected attributes,

Then create child class **Exhibition** that extends Event with the following attribute,

And create another child class **StageEvent** that extends Event with the following attribute,

Add suitable constructor (with super() if necessary) and getters/setters for the classes. Add method **projectedRevenue()** in parent and its child class.

Note: For Exhibition, each stall will produce Rs.10000 as revenue. For StageEvent, each seat produces Rs.50 revenue.

Input Format

Input consists of details of the event (Name, Detail, and ownerName).
The next input is the choice, 1 for Exhibition, and 2 for StageEvent).
If the input is 1, Enter the number of stalls. If the input is 2, enter the number of shows and the number of seats per show.
Refer sample input.

Output Format

Output prints the projected revenue of the event.

Sample Input

```
Science Fair
Explore Technology
ABCD
1
```

Sample Output

```
650000.0
```

Sample Input

```
Magic show
See magic without logic
SDFG
2
```

Sample Output

```
50000.0
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **A Multiplication Game**

John and Michael play the game of multiplication by multiplying an integer **p** by one of the numbers 2 to 9. John always starts with **p = 1** and multiply it by 1, and passes the result to Michael. Then, Michael multiplies the number by 2 and sends the result to John, then John multiplies by 3 and so on. Before a game starts, they draw an integer **N** and the winner is the one who first reaches $p \geq n$.

Create a class that has two functions:

1) A function to perform the multiplication operation

2) The main()

Input Format

The input consists of the value of n.

Output Format

The output prints the n value and who won the game separated by a space.
Refer to the sample output for formatting specifications.

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - MCQ

Q1

All of the above

Solution

No Solution

Q2

a string with num's type and hashcode

Solution

No Solution

Q3

Auto is a subclass of LandVehicle, and LandVehicle is a subclass of Vehicle.

Solution

No Solution

Q4

If a class extends another class that has only one constructor that takes a value, then the child class must explicitly declare at least one constructor.

Solution

No Solution

Q5

constructor

Solution

A constructor is a method that initializes an object immediately upon creation. It has the same name as that of class in which it resides.

Q6

Compilation error

Solution

No Solution

Q7

Program execute = true

Solution

No Solution

Q8

A

Solution

No Solution

Q9

A

Solution

No Solution

Q10

Compile time error

Solution

No Solution

Q11

All of the above.

Solution

No Solution

Q12

Its base classIts derived class

Solution

No Solution

Q13

It is the ability for a message/data to be processed in more than one form

Solution

No Solution

Q14

SimpleDateFormat

Solution

No Solution

Q15

All methods in an interface are automatically public.

Solution

No Solution

Q16

x

Solution

No Solution

Q17

a.y = 5

b.y = 6

a.x = 2

b.x = 2

IdentifyMyParts.x = 2

Solution

No Solution

Q18

a b

Solution

No Solution

Q19

Abstract class.

Solution

No Solution

Q20

prints "Cattle eats hay"

Solution

No Solution

Section 2 - CODING

Q1

Test Case

Input

```
12 24
```

Output

```
288
72
```

Weightage - 20

Input

```
123 421
```

Output

```
51783
1088
```

Weightage - 20

Input

Output

1254 1235	1548690 4978

Weightage - 20

Input **Output**

578 956	552568 3068

Weightage - 20

Input **Output**

486 684	332424 2340

Weightage - 20

Sample Input **Sample Output**

20 50	1000 140

Solution

```
import java.io.*;
import java.util.*;
class Rextangle {
    public static void main(String[] args) {
        int length,breadth;
        Scanner sc = new Scanner(System.in);
        length = sc.nextInt();
        breadth = sc.nextInt();
        System.out.println(length*breadth);
        System.out.println(2*(length+breadth));
    }
}
```

Q2

Test Case

Input **Output**

270 262	532

Weightage - 15

Input **Output**

133 457	590

Weightage - 15

Input

```
303
631
```

Output

```
934
```

Weightage - 10

Input

```
625
735
```

Output

```
1360
```

Weightage - 10

Input

```
629
199
```

Output

```
828
```

Weightage - 10

Input

```
797
687
```

Output

```
1484
```

Weightage - 10

Input

```
231
754
```

Output

```
985
```

Weightage - 10

Input

```
668
215
```

Output

```
883
```

Weightage - 5

Input

```
6000
0000
```

Output

```
6000
```

Weightage - 15

Sample Input

```
618
229
```

Sample Output

```
847
```

Solution

Header

```
import java.util.*;

class Main{

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();

        System.out.println(sum(a,b));
    }



static int sum(int a, int b){
    while (b != 0){
        int carry = a & b;
        a = a ^ b;
        b = carry << 1;
    }
return a;
}
```

Footer

```
}
```

Q3

Test Case

Input

```
25
```

Output

```
2 is a fibonacci number
2 is a power of two
3 is a fibonacci number
5 is a fibonacci number
```

Weightage - 20

Input

```
55
```

Output

```
2 is a fibonacci number
2 is a power of two
3 is a fibonacci number
5 is a fibonacci number
```

Weightage - 20

Input

Output

100	2 is a fibonacci number 2 is a power of two 3 is a fibonacci number 5 is a fibonacci number

Weightage - 20

Input **Output**

500	2 is a fibonacci number 2 is a power of two 3 is a fibonacci number 5 is a fibonacci number

Weightage - 20

Input **Output**

1200	2 is a fibonacci number 2 is a power of two 3 is a fibonacci number 5 is a fibonacci number

Weightage - 20

Sample Input **Sample Output**

5	2 is a fibonacci number 2 is a power of two 3 is a fibonacci number 5 is a fibonacci number

Sample Input **Sample Output**

15	2 is a fibonacci number 2 is a power of two 3 is a fibonacci number 5 is a fibonacci number

Solution

```
import java.io.*;
import java.util.*;
class fibAndPow {
    public static void Fibonacci(int n) {
    int i,a=2,b=3,c=0;
    System.out.println(a+" is a fibonacci number ");
    System.out.println(a+" is a power of two");
    System.out.println(b+" is a fibonacci number ");
    while(c<n) {
        c=a+b;
        if(c<=n) {
        System.out.println(c+" is a fibonacci number ");
        powerOfTwo(c);
        a=b;
        b=c;
        }
        else {
            break;
        }
    }
}
public static void powerOfTwo(int n) {
    if((int)(Math.ceil((Math.log(n) / Math.log(2)))) ==
   (int)(Math.floor(((Math.log(n) / Math.log(2)))))) {
    System.out.println(n+" is a power of two");
```

```
        }
        else {
             return;
        }
    }
    public static void main (String [] args) {
        int n;
        Scanner sc = new Scanner(System.in);
        n= sc.nextInt();
        Fibonacci(n);
    }
}
```

Q4

Test Case

Input

```
Science Fair
Explore Technology
ABCD
1
```

Output

```
400000.0
```

Weightage - 20

Input

```
Science Fair
Explore Technology
ABCD
1
```

Output

```
1000000.0
```

Weightage - 20

Input

```
Magic show
See magic without logic
SDFG
2
```

Output

```
150000.0
```

Weightage - 20

Input

```
Magic show
See magic without logic
SDFG
2
```

Output

```
250000.0
```

Weightage - 20

Input

```
Magic show
See magic without logic
SDFG
2
```

Output

```
125000.0
```

Weightage - 20

Sample Input

```
Science Fair
Explore Technology
```

Sample Output

```
650000.0
```

ABCD

Sample Input

```
Magic show
See magic without logic
SDFG
2
```

Sample Output

```
50000.0
```

Solution

```
import java.io.*;
import java.text.DecimalFormat;
import java.util.*;
class Event {
protected String name;
protected String detail;
protected String ownerName;
public Event(String name, String detail, String ownerName) {
    this.name = name;
    this.detail = detail;
    this.ownerName = ownerName;
}
public Event() {
    this.name = null;
    this.detail = null;
    this.ownerName = null;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getDetail() {
    return detail;
}
public void setDetail(String detail) {
    this.detail = detail;
}
public String getOwnerName() {
    return ownerName;
}
public void setOwnerName(String ownerName) {
    this.ownerName = ownerName;
}
public Double projectedRevenue() {
    return 0.0;
}
}
class Exhibition extends Event {
public int noOfStall;
public Exhibition(String name, String detail, String ownerName, int noOfStall) {
    super(name,detail,ownerName);
    this.noOfStall = noOfStall;
}
public Exhibition() {
    super();
    this.noOfStall = 0;
}

public int getNoOfStall() {
    return noOfStall;
}

public void setNoOfStall(int noOfStall) {
```

```
    this.noOfStall = noOfStall;
}
public Double projectedRevenue() {
    return (double) (noOfStall*10000);
}
}
class StageEvent extends Event {
public int noOfShows;
public int noOfSeatsPerShow;
public StageEvent(String name, String detail, String ownerName, int noOfShows,
        int noOfSeatsPerShow) {
    super(name,detail,ownerName);
    this.noOfShows = noOfShows;
    this.noOfSeatsPerShow = noOfSeatsPerShow;
}
public StageEvent(){
    super();
    this.noOfShows = 0;
    this.noOfSeatsPerShow = 0;
}
public int getNoOfShows() {
    return noOfShows;
}
public void setNoOfShows(int noOfShows) {
    this.noOfShows = noOfShows;
}
public int getNoOfSeatsPerShow() {
    return noOfSeatsPerShow;
}
public void setNoOfSeatsPerShow(int noOfSeatsPerShow) {
    this.noOfSeatsPerShow = noOfSeatsPerShow;
}
public Double projectedRevenue() {
    return (double) (50*noOfShows*noOfSeatsPerShow);
}
}
class Main {
    public static void main(String[] args) {
Event ev = new Event();
Scanner sc = new Scanner(System.in);
ev.name = sc.nextLine();
ev.detail = sc.nextLine();
ev.ownerName = sc.nextLine();
Event e1 = new Event(ev.name,ev.detail,ev.ownerName);
int n;
DecimalFormat dd = new DecimalFormat("0.0");
n = Integer.parseInt(sc.nextLine());
Exhibition ex = new Exhibition();
StageEvent se = new StageEvent();
if(n==1) {
    ex.noOfStall = Integer.parseInt(sc.nextLine());
    Exhibition ex1 = new Exhibition(ev.name,ev.detail,ev.ownerName,ex.noOfStall);
    double result = ex1.projectedRevenue();
    System.out.println(dd.format(result));
}
if(n==2) {
    se.noOfShows = Integer.parseInt(sc.nextLine());
    se.noOfSeatsPerShow = Integer.parseInt(sc.nextLine());
    StageEvent se1 = new StageEvent(ev.name,ev.detail,ev.ownerName,se.noOfShows,se.noOfSeatsPerShow);
    double result = se1.projectedRevenue();
    System.out.println(dd.format(result));
}
}
}
```

Q5

Test Case

Input

```
3000
```

Output

```
3000 John wins
```

Weightage - 20

Input

```
5550
```

Output

```
5550 Michael wins
```

Weightage - 20

Input

```
40500
```

Output

```
40500 John wins
```

Weightage - 20

Input

```
750
```

Output

```
750 John wins
```

Weightage - 20

Input

```
200
```

Output

```
200 Michael wins
```

Weightage - 20

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

Solution