IRC_JAVA_CA1_LE_SET1_Updated

Test Summary

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

Section 1 - CODING

Section Summary

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

Additional Instructions:

None

Q1. Write a program by creating a class Bicycle as a base class with a number of gears and speed of bicycle as integer attributes and create a class called MountainBike, a derived class that extends Bicycle class with an attribute seat height as an integer. Create a Test class to run the program and obtain the output in the console.
Note: Override toString() method to display the details of the bicycle.

Input Format

To get 3 integers from the user (Number of gears, Speed of bicycle, and Seat height).

Output Format

To display the desired output from the test class.

Constraints

integers only.

Sample Input

```
2 90 40
```

Sample Output

```
No of gears are 2
speed of bicycle is 90
seat height is 40
```

Sample Input

```
3 60 20
```

Sample Output

```
No of gears are 3
speed of bicycle is 60
seat height is 20
```

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

Q2. Java program to find the count of all digits of a number using class.
In this program, we will read a positive integer number and then calculate the count of all digits using a class.

Input Format

The input consists of a number.

Output Format

The output prints the count of all digits in the number.

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

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

Q3. 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

Q4. Write a Multiply function for two integers and use overload the function by changing the parameter for double data type.

Input Format

Input two Integers in a separate line & two Double in a separate line.

Output Format

One Integer value and Double value after performing multiplication in a separate line.

Sample Input

```
2
3
1.2
2.3
```

Sample Output

```
6
2.76
```

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

Q5. 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

Section 2 - MCQ

Section Summary

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

Additional Instructions:

None

Q1. Which of the following type of polymorphism in Java?

- Compile time polymorphism
- Execution time polymorphism

- Multiple polymorphism
- Multilevel polymorphism

Q2. Correct statement about a class and interface is/are

- An interface can extend multiple interfaces
- A class can implement multiple interfaces.
- Java support multiple inheritance using interfaces.
- All of the above.

Q3. Which of the following is not an OOPS concept in JAVA?

- Encapsulation
- Polymorphism
- Execution
- Abstraction

Q4. Public keyword is an

- identifier
- access specifire
- extention
- variable type

Q5. To enhance data security, all the variables of a class should be ideally declared as

- protected
- public
- private
- default

Q6. 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.

Q7. Consider the following code snippet:

This code is an example of using ____.

```
1 myImage.add(new Rectangle(10,10,10,10));
2
```

An anonymous class.

An anonymous object.

An abstract object.

An abstract class.

Q8. Which of the following is a method having same name as that of it’s class?

finalize

delete

class

constructor

Q9. A default constructor has

no argument and with return type

one argument

one argument and no return type

no argument and no return type

Q10. Which of these keywords is used to refer to member of base class from a subclass?

- this
- super
- upper
- None of the above

Q11. Which of the following statement is correct?

```
public class Vehicle
{
. . .
public void setVehicleAtrributes()
{
. . .
}
}
public class Auto extends Vehicle
{
. . .
public void setVehicleAtrributes()
{
. . .
}
}
```

- The subclass is shadowing a superclass method.
- The subclass is overloading a superclass method.
- The subclass is overriding a superclass method.
- This code will not compile.

Q12. Suppose the abstract class Message is defined below

A concrete subclass of Message, FrenchMessage, is defined. Which methods must FrenchMessage define?

```
1  public abstract class Message
2  {
3     private String value;
4     public Message(String initial)
5     {
6        value = initial;
7     }
8     public String getMessage()
9     {
10       return value;
11    }
12    public abstract String translate();
13 }
14
```

- translate() only
- getMessage() only
- The FrenchMessage constructor and translate() only

The FrenchMessage constructor, getMessage(), and translate()

Q13. Consider the following code snippet:

Assume that the Programmer class inherits from the Employee class, and both classes have an implementation of the increaseSalary method with the same set of parameters and the same return type. Which class's increaseSalary method is to be executed is determined by ____.

```
1  Employee anEmployee = new Programmer();
2  anEmployee.increaseSalary(2500);
3
```

the hierarchy of the classes.

the variable's type.

the actual object type.

it is not possible to determine which method is executed.

Q14. Consider the classes Parent and Child shown below and answer the question that follows.

What is the output of the following lines of code?

```
Child kid = new Child(-14);
Parent adult = new Parent();
System.out.println(kid.getValue() + " " + adult.getValue());
```

```
1  public class Parent
2  {
3     private int value = 100;
4     public int getValue()
5     {
6        return value;
7     }
8  }
9  public class Child extends Parent
10 {
11    private int value;
12    public Child(int number)
13    {
14       value = number;
15    }
16 }
17
```

100 100

-14 100

-14 -14

100 -14

Q15. What will be stored in the object obj in the following line of code ?
ItemType obj;

Memory address of allocated memory of object

NULL

Any arbitrary pointer

Garbage

Q16. What is Encapsulation?

Encapsulation is a technique to define different methods of same type

Encapsulation is the ability of an object to take on many forms

Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods

None of the mentioned options

Q17. While using parameterized constructor, how to specify the paramter list ?

No need to specify parameter list

Specify the parameter list as the same way it is specified in the method

Order of parameter list is not important

A constructor calls another constructor

Q18. Which among the following best defines single level inheritance?

A class inheriting a base class

A class inheriting a derived class

A class inheriting a nested class

A class which gets inherited by 2 classes

Q19. Which member of the superclass is never accessible to the subclass?

Public member

Protected member

Private member

None of the above

Q20. CompileTime Polymorphism can be achieved through

Up casting

Method Overloading

Method Onerriding

Inheriting

Answer Key & Solution

Section 1 - CODING

Q1

Test Case

Input

```
3 50 12
```

Output

```
No of gears are 3
speed of bicycle is 50
seat height is 12
```

Weightage - 10

Input

```
1 40 10
```

Output

```
No of gears are 1
speed of bicycle is 40
seat height is 10
```

Weightage - 10

Input

```
3 50 28
```

Output

```
No of gears are 3
speed of bicycle is 50
seat height is 28
```

Weightage - 10

Input

```
1 44 22
```

Output

```
No of gears are 1
speed of bicycle is 44
seat height is 22
```

Weightage - 10

Input

```
4 60 30
```

Output

```
No of gears are 4
speed of bicycle is 60
seat height is 30
```

Weightage - 10

Input

```
2 69 30
```

Output

```
No of gears are 2
speed of bicycle is 69
seat height is 30
```

Weightage - 10

Input

```
4 80 22
```

Output

```
No of gears are 4
speed of bicycle is 80
seat height is 22
```

Weightage - 20

Input

```
1 79 43
```

Output

```
No of gears are 1
speed of bicycle is 79
seat height is 43
```

Weightage - 20

Sample Input

```
2 90 40
```

Sample Output

```
No of gears are 2
speed of bicycle is 90
seat height is 40
```

Sample Input

```
3 60 20
```

Sample Output

```
No of gears are 3
speed of bicycle is 60
seat height is 20
```

Solution

```
import java.util.Scanner;
class Bicycle
{
    public int gear;
    public int speed;

    public Bicycle(int gear, int speed)
    {
        this.gear = gear;
        this.speed = speed;
    }

    public void applyBrake(int decrement)
    {
        speed -= decrement;
    }

    public void speedUp(int increment)
    {
        speed += increment;
    }

    // toString() method to print info of Bicycle
    public String toString()
    {
        return("No of gears are "+gear
                +"\n"
                + "speed of bicycle is "+speed);
    }
}

class MountainBike extends Bicycle
{

    public int seatHeight;

    public MountainBike(int gear,int speed,
```

```
                        int startHeight)
    {
        super(gear, speed);
        seatHeight = startHeight;
    }

    public void setHeight(int newValue)
    {
        seatHeight = newValue;
    }

    @Override
    public String toString()
    {
        return (super.toString()+
                "\nseat height is "+seatHeight);
    }

}

class Test
{
    public static void main(String args[])
    {
         int gear,speed,startHeight;
         Scanner sc=new Scanner(System.in);
         gear=sc.nextInt();
         speed=sc.nextInt();
         startHeight=sc.nextInt();
        MountainBike mb = new MountainBike(gear,speed,startHeight);
        System.out.println(mb.toString());

    }
}
```

Q2

Test Case

Input

```
2147483647
```

Output

```
Count of all digits: 10
```

Weightage - 15

Input

```
222222888
```

Output

```
Count of all digits: 9
```

Weightage - 15

Input

```
77777777
```

Output

```
Count of all digits: 8
```

Weightage - 15

Input

```
1234567
```

Output

```
Count of all digits: 7
```

Weightage - 15

Input

```
654321
```

Output

```
Count of all digits: 6
```

Weightage - 15

Input

```
3214
```

Output

```
Count of all digits: 4
```

Weightage - 15

Input

```
12345
```

Output

```
Count of all digits: 5
```

Weightage - 10

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

Solution

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int count = 0, num;
        Scanner s = new Scanner(System.in);
        num = s.nextInt();
        while(num != 0)
        {
            num /= 10;
            ++count;
        }
        System.out.println("Count of all digits: " + count);
```

```
    }
}
```

Q3

Test Case

Input

```
12  24
```

Output

```
288
72
```

Weightage - 20

Input

```
123  421
```

Output

```
51783
1088
```

Weightage - 20

Input

```
1254  1235
```

Output

```
1548690
4978
```

Weightage - 20

Input

```
578  956
```

Output

```
552568
3068
```

Weightage - 20

Input

```
486  684
```

Output

```
332424
2340
```

Weightage - 20

Sample Input

```
20  50
```

Sample Output

```
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));
    }
}
```

Q4

Test Case

Input	Output
12 23 1.32 4.21	276 5.5572

Weightage - 50

Input	Output
76 21 3.211 43.21	1596 138.74731

Weightage - 50

Sample Input

Sample Output

2 3 1.2 2.3	6 2.76

Solution

Header

```
import java.util.*;
class Main {

    int Multiply(int a, int b)
    {
        return a * b;
    }

    double Multiply(double a, double b)
    {
        return a * b;
    }
```

Footer

```
public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);
        int a=in.nextInt();
        int b=in.nextInt();
        double c = in.nextDouble();
        double d=in.nextDouble();
```

```
        Main m=new Main();
        System.out.println(m.Multiply(a, b));
        System.out.println(m.Multiply(c, d));
    }
}
```

Q5

Test Case

Input

```
270
262
```

Output

```
532
```

Weightage - 15

Input

```
133
457
```

Output

```
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

Output

231 754	985

Weightage - 10

Input	Output
668 215	883

Weightage - 5

Input	Output
6000 0000	6000

Weightage - 15

Sample Input	Sample Output
618 229	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

```
}
```

Section 2 - MCQ

Q1

Compile time polymorphism

Solution

No Solution

Q2

All of the above.

Solution

No Solution

Q3

Execution

Solution

No Solution

Q4

access specifire

Solution

No Solution

Q5

private

Solution

No Solution

Q6

All of the above.

Solution

No Solution

Q7

An anonymous object.

Solution

No Solution

Q8

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.

Q9 no argument and no return type

Solution

No Solution

Q10 super

Solution

No Solution

Q11 The subclass is overriding a superclass method.

Solution

No Solution

Q12 translate() only

Solution

No Solution

Q13 the actual object type.

Solution

No Solution

Q14 100 100

Solution

No Solution

Q15 NULL

Solution

No Solution

Q16 Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods

Solution

No Solution

Q17 Specify the parameter list as the same way it is specified in the method

Solution

No Solution

Q18 A class inheriting a base class

Solution

No Solution

Q19 Private member

Solution

No Solution

Q20 Method Overloading

Solution

No Solution