OOPS - 1

Correct Statement

Send Feedback

Select the correct statement(s).

Options

This problem may have one or more correct answers

- [x] OOPS refers to using objects in programming ✔
- [x] A class is a user defined blueprint from which objects are created. ✔
- [ ] Object of same class have different properties.
- [x] Object is an instance of a class. ✔

Hurray! Correct Answer

Object Creation

Send Feedback

Which of the following method can be used to create an object of student class?

Options

This problem has only one correct answer

- ( ) Student s1=new Student;
- ( ) Student s1;
- (•) Student s1=new Student();
- ( ) Student s1=" ";

Hurray! Correct Answer

Predict the output

Send Feedback

What would be the output of the following code?

```
class Student{
   int roll_number;
   String name;
}
class DPS {
   public static void main (String[] args) {
      Student s=new Student();
      s.roll_number=5;
      s.name="Rohit";
      System.out.println(s.roll_number+" "+s.name);
   }
}
```

Options

This problem has only one correct answer

- (•) 5 Rohit
- ( ) Rohit 5
- ( ) error

Hurray! Correct Answer

Find the error

Send Feedback

Which line of the following code would give an error?

```
class Car{
   int year;
   String company_name;
}
class New_Car {
   public static void main (String[] args) {
      Car c=new Car();       //Line 1
      year=2018;             //Line 2
      c.company_name="Honda"; //Line 3
   }
}
```

Options

This problem has only one correct answer

- Line 1
- Line 2 ✓
- Line 3
- No error

Hurray! Correct Answer

Solution Description

In line 2 , the year is not a variable in main.So we need to use the object of class Car.

Possible output

Send Feedback

What can be the possible output of the following code?

```
class Student{
   int roll_no;
   String name;
}
class Test {
      public static void main (String[] args) {
      Student s=new Student();
      System.out.println(s);
   }
}
```

Options

This problem has only one correct answer

- 547a85bc
- [I@232204a1
- Student@232204a1 ✓
- Student@4578ig32

Hurray! Correct Answer

Solution Description

The class is of Type student and it would be followed by a hexadecimal code.
So the address would be like Student@.....
hexadecimal code.
(Option 4 is not a hexadecimal code.)

Access Modifiers

Send Feedback

Which access modifier can be used to access a variable outside the class and within the package?

Options

This problem may have one or more correct answers

- [x] Public ✓
- [ ] Private
- [x] Default ✓

Hurray! Correct Answer

Check for error

Send Feedback

Would the following code generate any error?

```
class Student{
   private int roll_no;
   String name;
}
class DPS {
      public static void main (String[] args) {
      Student s=new Student();
      s.name="Neha";
      System.out.println(s.name);
   }
}
```

Options

This problem has only one correct answer

- ○ Yes
- ◉ No

Hurray! Correct Answer

Predict the output

Send Feedback

What would be the output of the following code?

```
class Mobile{
   private int year;
   String company_name;
}
class new_device {
   public static void main (String[] args) {
      Mobile c=new Mobile();
      c.year=2018;
      c.company_name="Apple";
      System.out.println(c.company_name);
   }
}
```

Options

This problem has only one correct answer

- ○ Apple
- ○ No output
- ◉ Error

Hurray! Correct Answer

Solution Description

The variable year in Mobile class is private.
So it cannot be accessed outside the class.

Hence the line, c.year=2018, would give an error.

Predict the output

Send Feedback

What will be the output of the following code?(Considering both the classes are in the same package).

```
class Employee{
   String name;
      private int emp_id;
   public void set_id(int id)
   {
      if(id>0)
      emp_id=id;
      else
      System.out.println("Invalid id");
   }
   public int get_id()
   {
      return emp_id;
   }
}
class office {
   public static void main (String[] args) {
      Employee e=new Employee();
      e.set_id(10);
      System.out.println(e.get_id());
   }
}
```

Options

This problem has only one correct answer

- (●) 10
- ( ) 0
- ( ) error

Hurray! Correct Answer

Fill the output

Send Feedback

Fill the output of the following code.(Considering both the classes are in the same package.)

```
class Employee{
   String name;
      private int emp_id;
   public void set_id(int id)
   {
      if(id>0)
      emp_id=id;
      else
      System.out.println("Invalid id");
   }
   public int get_id()
   {
      return emp_id;
   }
   public void set_name(String n)
   {
      name=n;
   }
}
class Office {
   public static void main (String[] args) {
      Employee e=new Employee();
      e.set_id(10);
      e.set_name("Naman");
      System.out.println(e.get_id()+" "+e.name);
   }
}
```

Answer

10 Naman ✔

Correct Answer

What will be the output of the following code?

```
class Test
   {
      int a;
      int b;

   public void set(int a, int b)
   {
      b = a;
      this.b = b;
   }

   void display()
   {
      System.out.println("a=" + a + " b=" + b);
   }
}
class T{
   public static void main(String[] args)
   {
      Test object = new Test();
      object.set(10,20);
      object.display();
   }
}
```

a=0 b=10 ✔

Correct Answer

Solution Description

When object.set function is called, the local variables are a and b which are passed on to the function as parameters.

a=10,b=20.
But a has been assigned to b. That means b comes 10.
Now this.b=b sets the variable b of class Test equal to 10 and a remains 0.

Keyword

Send Feedback

Which keyword is a reference variable that refers to the current object?

Options

This problem has only one correct answer

- ○ object
- ◉ this
- ○ class
- ○ private

✔ Hurray! Correct Answer

Predict the output

Send Feedback

What will be the output of the following code?

```
class Ninja
{
   Ninja(String name)
   {
      System.out.println("Constructor one " + name);
   }
   Ninja(String name, int age)
   {
      System.out.println("Constructor two " + name + "
"+ age);
   }
   Ninja(long id)
   {
      System.out.println("Constructor three " + id);
   }
}

class Student
{
   public static void main(String[] args)
   {
      Ninja geek3 = new Ninja("Dharmesh", 26);
   }
}
```

Options

This problem has only one correct answer

- ○ Constructor one Dharmesh
- ◉ Constructor two Dharmesh 26
- ○ Constructor three 26

Hurray! Correct Answer

Solution Description

The constructor with two arguments string and integer is called.

Predict the output

Send Feedback

What will be the output of the following code?

```
class Test
{
   int a;
   int b;
   Test()
   {
      this(10, 20);
      System.out.print("constructor one ");
   }
   Test(int a, int b)
   {
      this.a = a;
      this.b = b;
      System.out.print("constructor two ");
   }
}
class new_test{
   public static void main(String[] args)
   {
      Test object = new Test();
   }
}
```

Options

Attempts left: 0/2

This problem has only one correct answer

- ○ constructor one
- ○ constructor two
- ◉ constructor two constructor one
- ○ constructor one constructor two

Hurray! Correct Answer

Solution Description

When the object of Test class is created,the constructor with no arguments is called.
When this(10,20) is encountered , the constructor with two arguments int and int is called because this is the reference of the object. So using this way we can call more than one constructor.

Final Keyword

Send Feedback

Select the correct statement(s) about final keyword?

Options

Attempts left: 1/2

This problem may have one or more correct answers

- [x] Final variable can be initialized only once and cannot be modified further. ✔
- [ ] We can initialize final variable inside a function.
- [x] We can initialize final variable inside constructor. ✔
- [x] Final keyword is a non-access modifier. ✔

Hurray! Correct Answer

Predict the output

Send Feedback

What will be the output of the following code?

```
class Pen{
   final int price = 15;
}

public class MCQs {
   public static void main(String[] args) {
      Pen p = new Pen();
      p.price = 20;
      System.out.println(p.price);
   }
}
```

Options

This problem has only one correct answer

- ( ) 15
- ( ) 20
- (•) Error

Hurray! Correct Answer

Solution Description

Since the variable price is final so we cannot update its value.

Predict the output

Send Feedback

What will be the output of the following code?

```
class Car{
   static int year;
   String company_name;
}
class new_Car {
   public static void main (String[] args) {
      Car c=new Car();
      Car.year=2018;
      c.company_name="KIA";
      Car d=new Car();
      System.out.print(d.year);
   }
}
```

Options

This problem has only one correct answer

- ( ) 0
- ( ) Error
- (•) 2018

Hurray! Correct Answer

Solution Description

Year is of static type i.e. only one memory block of year is created , same for every object. So, it would print 2018.

Find the error

Send Feedback

Which line of the following code would generate an error?

```
class Test{
   static int marks;
   void set_marks(int marks)
   {
      this.marks=marks;            //Line 1
   }
}
class MCQ {
   public static void main (String[] args) {
      Test t=new Test();
      t.set_marks(78);             //Line 2
      System.out.print(Test.marks);   //Line 3
   }
}
```

Attempts left: 1/2

Options

This problem has only one correct answer

- ○ Line 1
- ○ Line 2
- ○ Line 3
- ◉ No error

Hurray! Correct Answer

Solution Description

There is no error in this code. In this class, there is just a single variable, which is, static variable. Static variables are property of the class, but you can access it through objects. Hence, line 1 and 2 did not generate any error.

Predict the output

Send Feedback

What would be the output of the following code?

```
class Test
{
   static int a = 10;
   static int b;
   static void fun(){
      b = a * 4;
   }
}
class MCQ{
  public static void main(String[] args)
   {
     Test t=new Test();
     t.fun();
     System.out.print(Test.a+Test.b);
   }
}
```

Options

This problem has only one correct answer

- ○ 10
- ○ 20
- ◉ 50
- ○ Error

Hurray! Correct Answer

Solution Description

When t.fun() is called . a=10 and using this b=40.
We print a+b=50.

Find the error

Send Feedback

Which of the following line(s)would produce an error?

```
class Test
{
   static int a = 10;
   int b = 20;
   static void fun1()
   {
      a = 20;                    //Line 1
      b = 10;                    //Line 2
      fun2();                    //Line 3
      System.out.print(this.b);        //Line 4
   }
   void fun2()
   {
      System.out.println("from m2");
   }
}
class MCQ{
   public static void main(String[] args)
   {
      Test.fun1();
   }
}
```

Options

This problem may have one or more correct answers

- ☐ Line 1
- ☑ Line 2 ✔
- ☑ Line 3 ✔
- ☑ Line 4 ✔

Hurray! Correct Answer

Solution Description

Line 2: static functions cannot use non-static variables.

Line 3: Static function cannot call non-static function.

Line 4:this or super keyword cannot be used inside a static function.

Complex Number Problem

Send Feedback

A ComplexNumber class contains two data members : one is the real part (R) and the other is imaginary (I) (both integers).
Implement the Complex numbers class that contains following functions -
1. constructor
You need to create the appropriate constructor.
2. plus -
This function adds two given complex numbers and updates the first complex number.
e.g.
if C1 = 4 + i5 and C2 = 3 +i1
C1.plus(C2) results in:
C1 = 7 + i6 and C2 = 3 + i1
3. multiply -
This function multiplies two given complex numbers and updates the first complex number.
e.g.
if C1 = 4 + i5 and C2 = 1 + i2
C1.multiply(C2) results in:
C1 = -6 + i13 and C2 = 1 + i2
4. print -
This function prints the given complex number in the following format :
a + ib
Note : There is space before and after '+' (plus sign) and no space between 'i' (iota symbol) and b.
Input Format :
Line 1 : Two integers - real and imaginary part of 1st complex number
Line 2 : Two integers - real and imaginary part of 2nd complex number
Line 3 : An integer representing choice (1 or 2) (1 represents plus function will be called and 2 represents multiply function will be called)
Output format :
Check details of 'print' function given above.
Sample Input 1 :
4 5
6 7
1
Sample Output 1 :
10 + i12
Sample Input 2 :
4 5
6 7
2
Sample Output 2 :
-11 + i58

```
public class ComplexNumbers {
    // Complete this class
    private int real;
    private int imaginary;

    public ComplexNumbers(int real, int imaginary){
        this.real = real;
        this.imaginary = imaginary;
```

```
    }

    public void print(){
        System.out.println(real+" + i"+imaginary);
    }

    public void plus(ComplexNumbers c){
        this.real = this.real+c.real;
        this.imaginary = this.imaginary+c.imaginary;
    }

    public void multiply(ComplexNumbers c){
        int temp = (this.real*c.real)-(this.imaginary*c.imaginary);
        this.imaginary = (this.real*c.imaginary)+(this.imaginary*c.real);
        this.real = temp;
    }
}
```

Polynomial Class Problem

Send Feedback

Implement a polynomial class, that contains following functions -

1. setCoefficient -

This function sets coefficient for a particular degree value. If term with given degree is not there in the polynomial, then corresponding term (with specified degree and value) is added. If the term is already present in the polynomial, then previous coefficient value is replaced by given coefficient value.

2. add -

Adds two polynomials and returns a new polynomial which has the result.

3. subtract -

Subtracts two polynomials and returns a new polynomial which has the result.

4. multiply -

Multiplies two polynomials and returns a new polynomial which has the result.

5. print -

Prints all the terms (only terms with non zero coefficients are to be printed) in increasing order of degree.

Print pattern for a single term : "x"

And multiple terms should be printed separated by space. For more clarity, refer sample test cases.

Note : Only keep those terms which have non - zero coefficients.

Input Format:

The first line of input contains count of the number of coefficients in polynomial 1(C1)

The next line of input has C1 degrees for polynomial 1.

The next line of input has C1 coefficients for polynomial 1.

The next line of input contains count of the number of coefficients in polynomial 2(C2)

The next line of input has C2 degrees for polynomial 2.

```
The next line of input has C2 coefficients for polynomial 2.
The next line of input has the choice for the function you want to
implement.
Output Format:
The output will be printed in case of print function same as that of print
function format.
Sample Input 1 :
P1 : 1x2 2x3 4x6
P2 : 3x4 1x2
Sample Output 1 :
P1 + P2 = 2x2 2x3 3x4 4x6
Sample Input 2 :
P1 : 1x2 2x3 4x6
P2 : 3x4 1x2
Sample Output 2 :
P1 - P2 = 2x3 -3x4 4x6

public class Polynomial {


   /* This function sets coefficient for a particular degree value, if
degree is not there in the polynomial
    *  then corresponding term(with specified degree and value is added
int the polynomial. If the degree
    *  is already present in the polynomial then previous coefficient is
replaced by
    *  new coefficient value passed as function argument
   */

   int polynomial[] = new int[5];

   public void setCoefficient(int degree, int coeff){
       if(degree>=polynomial.length)
           growArray(degree+1);
       polynomial[degree] = coeff;

   }

   public void growArray(int newSize){
       int temp[] = new int[polynomial.length];
       for(int i=0; i<temp.length; i++){
           temp[i] = polynomial[i];
       }
       polynomial = new int[newSize];
       for(int i=0; i<temp.length; i++){
           polynomial[i] = temp[i];
       }
   }

   // Prints all the terms(only terms with non zero coefficients are to be
printed) in increasing order of degree.
   public void print(){

       for(int i=0; i<polynomial.length; i++){
```

```
            if(polynomial[i]!=0)
                System.out.print(polynomial[i]+"x"+i+" ");
        }

    }


    // Adds two polynomials and returns a new polynomial which has result
    public Polynomial add(Polynomial p){

        Polynomial ans = new Polynomial();
        int minlen =
this.polynomial.length>p.polynomial.length?p.polynomial.length:this.polyno
mial.length;
        int commonIndex=0;

        for(int i=0; i<minlen; i++){
            if(p.polynomial[i]!=0 || this.polynomial[i]!=0)
                ans.setCoefficient(i,this.polynomial[i]+p.polynomial[i]);
            commonIndex = i;
        }
        for(int i=commonIndex; i<p.polynomial.length; i++)
            ans.setCoefficient(i,p.polynomial[i]);
        for(int i=commonIndex; i<this.polynomial.length; i++)
            ans.setCoefficient(i,this.polynomial[i]);

        return ans;
    }

    // Subtracts two polynomials and returns a new polynomial which has
result
    public Polynomial subtract(Polynomial p){

        Polynomial ans = new Polynomial();
        int minlen =
this.polynomial.length>p.polynomial.length?p.polynomial.length:this.polyno
mial.length;
        int commonIndex=0;

        for(int i=0; i<minlen; i++){
            if(i>=ans.polynomial.length)
                    ans.growArray(i*2);
            if(p.polynomial[i]!=0 && this.polynomial[i]!=0)
                ans.setCoefficient(i,this.polynomial[i]-p.polynomial[i]);
        }

        for(int i=0; i<p.polynomial.length; i++){
            if(i>=ans.polynomial.length)
                    ans.growArray(i*2);
            if(ans.polynomial[i] == 0)
                ans.setCoefficient(i,0-p.polynomial[i]);
        }

        for(int i=0; i<this.polynomial.length; i++){
```

```
            if(i>=ans.polynomial.length)
                    ans.growArray(i*2);
            if(ans.polynomial[i] == 0)
                ans.setCoefficient(i,this.polynomial[i]);
        }

        return ans;
    }

    // Multiply two polynomials and returns a new polynomial which has
result
    public Polynomial multiply(Polynomial p){

        Polynomial ans = new Polynomial();


        for(int i=0; i<this.polynomial.length; i++){
            for(int j=0; j<p.polynomial.length; j++){
                if(i+j>=ans.polynomial.length)
                    ans.growArray(i+j*2);
                ans.polynomial[i+j] += this.polynomial[i]*p.polynomial[j];
            }
        }
        // for(int i=0; i<ans.polynomial.length; i++){
        //     ans.setCoefficient(i, temp[i]);
        // }

        return ans;
    }

}
```