Java Test

Java Test

* Required

Email address *

Your email

```
1. class San
2. {
3.  public void m1 (int i,float f)
4.  {
5.   System.out.println(" int float method");
6.  }
7.
8.  public void m1(float f,int i);
9.   {
10.  System.out.println("float int method");
11.  }
12.
13.  public static void main(String[]args)
14.  {
15.    San s=new San();
16.        s.m1(20,20);
17.  }
18. }
```

- ◯ a) int float method
- ◯ b) float int method
- ◯ c) compile time error
- ◯ d) run time error

```
1.  class static_out
2.  {
3.      static int x;
4.      static int y;
5.      void add(int a, int b)
6.      {
7.          x = a + b;
8.          y = x + b;
9.      }
10. }
11. class static_use
12. {
13.     public static void main(String args[])
14.     {
15.         static_out obj1 = new static_out();
16.         static_out obj2 = new static_out();
17.         int a = 2;
18.         obj1.add(a, a + 1);
19.         obj2.add(5, a);
20.         System.out.println(obj1.x + " " + obj2.y);
21.     }
22. }
```

- ○ a) 7 7
- ○ b) 6 6
- ○ c) 7 9
- ○ d) 9 7

```
1.  class box
2.  {
3.      int width;
4.      int height;
5.      int length;
6.      int volume;
7.      void finalize()
8.      {
9.          volume = width*height*length;
10.         System.out.println(volume);
11.     }
12.     protected void volume()
13.    {
14.         volume = width*height*length;
15.         System.out.println(volume);
16.    }
17. }
18. class Output
19. {
20.     public static void main(String args[])
21.     {
22.         box obj = new box();
23.         obj.width=5;
24.         obj.height=5;
25.         obj.length=6;
26.         obj.volume();
27.     }
28. }
```

- a) 150
- b) 200
- c) Run time error
- d) Compilation error

```
1.  class A
2.  {
3.      public int i;
4.      protected int j;
5.  }
6.  class B extends A
7.  {
8.      int j;
9.      void display()
10.     {
11.         super.j = 3;
12.         System.out.println(i + " " + j);
13.     }
14. }
15. class Output
16. {
17.     public static void main(String args[])
18.     {
19.         B obj = new B();
20.         obj.i=1;
21.         obj.j=2;
22.         obj.display();
23.     }
24. }
```

- a) 1 2
- b) 2 1
- c) 1 3
- d) 3 1

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.           String x = Boolean.toString(false);
6.        }
7.    }
```

- a) True
- b) False
- c) System Dependent
- d) Compilation Error

- Option 1

```
1.     class overload
2.     {
3.         int x;
4.         int y;
5.         void add(int a)
6.         {
7.             x =  a + 1;
8.         }
9.         void add(int a, int b)
10.        {
11.            x =  a + 2;
12.        }
13.    }
14.    class Overload_methods
15.    {
16.        public static void main(String args[])
17.        {
18.            overload obj = new overload();
19.            int a = 0;
20.            obj.add(6);
21.            System.out.println(obj.x);
22.        }
23.    }
```

- a) 5
- b) 6
- c) 7
- d) 8

```
1.  class overload
2.  {
3.      int x;
4.      int y;
5.      void add(int a)
6.      {
7.          x =  a + 1;
8.      }
9.      void add(int a , int b)
10.     {
11.         x =  a + 2;
12.     }
13. }
14. class Overload_methods
15. {
16.     public static void main(String args[])
17.     {
18.         overload obj = new overload();
19.         int a = 0;
20.         obj.add(6, 7);
21.         System.out.println(obj.x);
22.     }
23. }
```

- a)6
- b) 7
- c) 8
- d) 9on 1

```
1.  class test
2.  {
3.      int a;
4.      int b;
5.      void meth(int i , int j)
6.      {
7.          i *= 2;
8.          j /= 2;
9.      }
10. }
11. class Output
12. {
13.     public static void main(String args[])
14.     {
15.         test obj = new test();
16.         int a = 10;
17.         int b = 20;
18.         obj.meth(a , b);
19.         System.out.println(a + " " + b);
20.     }
21. }
```

- a) 10 20
- b) 20 10
- c) 20 40
- d) 40 20

```
1.    class access
2.    {
3.        public int x;
4.        private int y;
5.        void cal(int a, int b)
6.        {
7.            x =  a + 1;
8.            y =  b;
9.        }
10.   }
11.   class access_specifier
12.   {
13.       public static void main(String args[])
14.       {
15.           access obj = new access();
16.           obj.cal(2, 3);
17.           System.out.println(obj.x + " " + obj.y);
18.       }
19.   }
```

- a) 3 3
- b) 2 3
- c) Runtime Error
- d) Compilation Error

Question 1

Question 1

```
package main;

class Base {
    public void Print() {
        System.out.println("Base");
    }
}

class Derived extends Base {
    public void Print() {
        System.out.println("Derived");
    }
}

class Main{
    public static void DoPrint( Base o ) {
        o.Print();
    }
    public static void main(String[] args) {
        Base x = new Base();
        Base y = new Derived();
        Derived z = new Derived();
        DoPrint(x);
        DoPrint(y);
        DoPrint(z);
    }
}
```

- ◯ Base, Derived, Derived
- ◯ Derived,Base, Derived
- ◯ Derived, Derived,Base
- ◯ Compilation Error

Question 2

Question 2

```
package main;

// filename Main.java
class Point {
    protected int x, y;

    public Point(int _x, int _y) {
        x = _x;
        y = _y;
    }
}

public class Main {
    public static void main(String args[]) {
      Point p = new Point();
      System.out.println("x = " + p.x + ", y = " + p.y);
    }
}
```

Your answer

Question 3

```
class First
{
    int i = 10;

    public First(int j)
    {
        System.out.println(i);
        this.i = j * 10;
    }
}

class Second extends First
{
    public Second(int j)
    {
        super(j);
        System.out.println(i);
        this.i = j * 20;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Second n = new Second(20);
        System.out.println(n.i);
    }
}
```

- ◯ 200 10 400
- ◯ 400 200 10
- ◯ 10 200 400
- ◯ None of the above

Question 4

```
import java.util.*;
class I
{
    public static void main (String[] args)
    {
        Object i = new ArrayList().iterator();
        System.out.print((i instanceof List) + ", ");
        System.out.print((i instanceof Iterator) + ", ");
        System.out.print(i instanceof ListIterator);
    }
}
```

- ◯ true,false, false
- ◯ false, false, true
- ◯ false, true, false
- ◯ compilation Error

Question 5

```
public class Calculator
{
    int num = 100;
    public void calc(int num)  { this.num = num * 10;  }
    public void printNum()     { System.out.println(num); }

    public static void main(String[] args)
    {
        Calculator obj = new Calculator();
        obj.calc(2);
        obj.printNum();
    }
}
```

- ◯ A) 20
- ◯ B) 100
- ◯ C) 1000
- ◯ D) 2

Question 6

```
public class MyStuff
{
    String name;

    MyStuff(String n) {  name = n;  }

    public static void main(String[] args)
    {
        MyStuff m1 = new MyStuff("guitar");
        MyStuff m2 = new MyStuff("tv");
        System.out.println(m2.equals(m1));
    }

    @Override
    public boolean equals(Object obj)
    {
        MyStuff m = (MyStuff) obj;
        if (m.name != null)  { return true;  }
        return false;
    }
}
```

- A) The output is true and MyStuff fulfills the Object.equals() contract.
- B) The output is false and MyStuff fulfills the Object.equals() contract.
- C) The output is true and MyStuff does NOT fulfill the Object.equals() contract.
- D) The output is false and MyStuff does NOT fulfill the Object.equals() contract.

Question 7

```
class Alpha
{
    public String type = "a ";
    public Alpha() {  System.out.print("alpha "); }
}

public class Beta extends Alpha
{
    public Beta()  {  System.out.print("beta ");  }

    void go()
    {
        type = "b ";
        System.out.print(this.type + super.type);
    }

    public static void main(String[] args)
    {
        new Beta().go();
    }
}
```

- A) alpha beta b b
- B) alpha beta a b
- C) beta alpha b b
- D) beta alpha a b

Question 8

```
public class Test
{
    public static void main(String[] args)
    {
        StringBuilder s1 = new StringBuilder("Java");
        String s2 = "Love";
        s1.append(s2);
        s1.substring(4);
        int foundAt = s1.indexOf(s2);
        System.out.println(foundAt);
    }
}
```

- A) -1
- B) 3
- C) 4
- D) A StringIndexOutOfBoundsException is thrown at runtime.

Question 9

```
class Writer
{
    public  static void write()
    {
        System.out.println("Writing...");
    }
}
class Author extends Writer
{
    public  static void write()
    {
        System.out.println("Writing book");
    }
}

public class Programmer extends Author
{
    public  static void write()
    {
        System.out.println("Writing code");
    }

    public static void main(String[] args)
    {
        Author a = new Programmer();
        a.write();
    }
}
```

- A) Writing…
- B) Writing book
- C) Writing code
- D) Compilation fails

Question 10

```
class Person
{
    private void who()
    {
        System.out.println("Inside private method Person(who)");
    }

    public static void whoAmI()
    {
        System.out.println("Inside static method, Person(whoAmI)");
    }

    public void whoAreYou()
    {
        who();
        System.out.println("Inside virtual method, Person(whoAreYou)");
    }
}

class Kid extends Person
{
    private void who()
    {
        System.out.println("Kid(who)");
    }

    public static void whoAmI()
    {
        System.out.println("Kid(whoAmI)");
    }

    public void whoAreYou()
    {
        who();
        System.out.println("Kid(whoAreYou)");
    }
}
public class Gfg
{
    public static void main(String args[])
    {
        Person p = new Kid();
        p.whoAmI();
        p.whoAreYou();
    }
}
```

- ◯ Inside static method, People(whoAmI) Kid(who) Kid(whoAreYou)
- ◯ Kid(whoAreYou) Kid(who) Inside static method, People(whoAmI)
- ◯ Inside static method, People(whoAmI) Kid(whoAreYou) Kid(who)
- ◯ Compilation Error

Send me a copy of my responses.

SUBMIT

Never submit passwords through Google Forms.