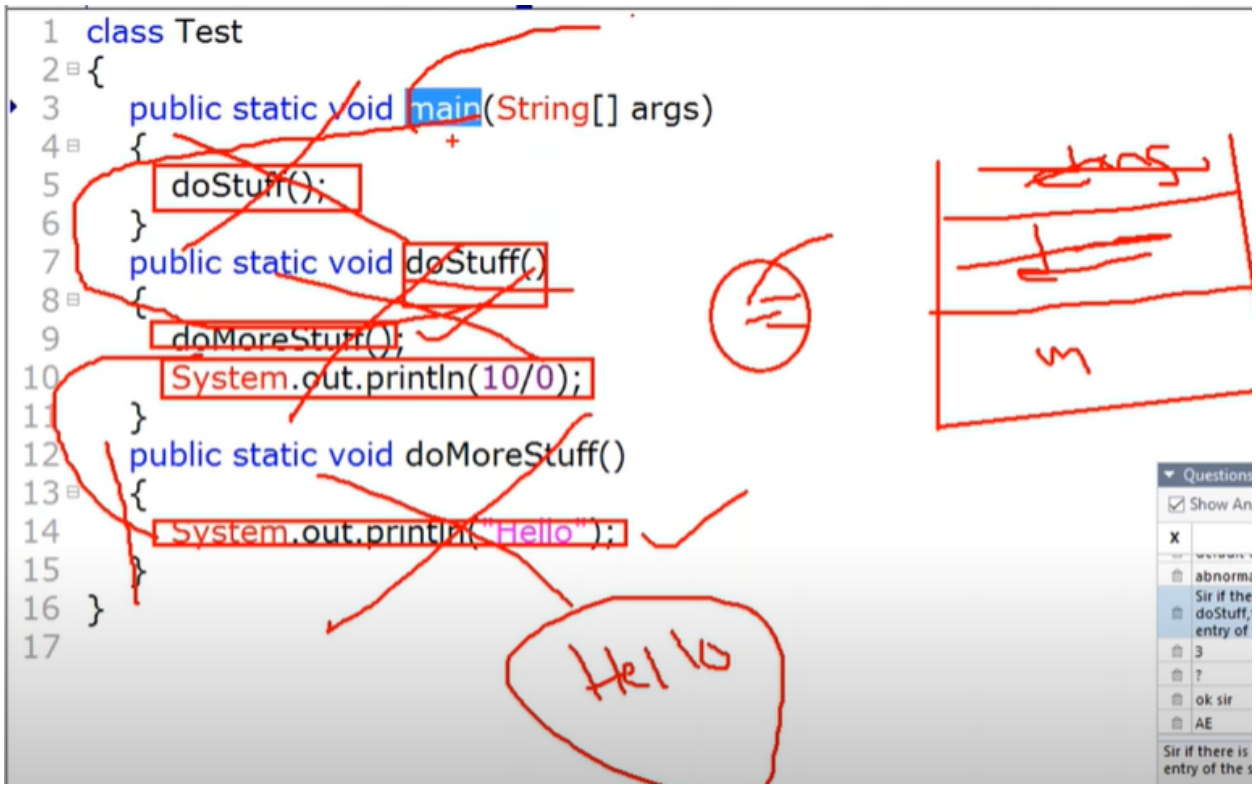
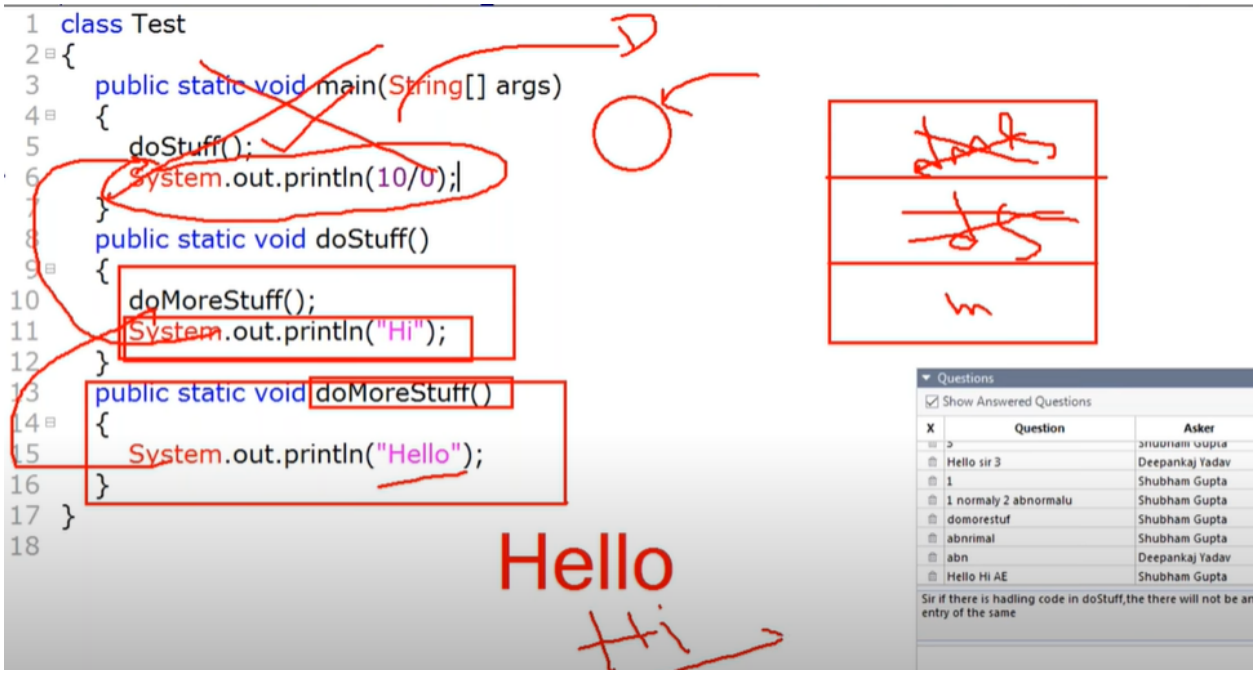
```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          doStuff();
6      }
7      public static void doStuff()
8      {
9          doMoreStuff();
10     }
11     public static void doMoreStuff()
12     {
13         System.out.println("Hello");
14     }
15 }
16
```

Stack representation of normally terminated code

```
Exception in thread main java.lang.AE: / by zero
                    at Test.doMoreStuff()
                    at.Test.doStuff()
                    at Test.main()
```




```
Hello
Exception in thread main java.lang.AE: / by zero
                    at.Test.doStuff()
                    at Test.main()
```




```
Hello
Hi
Exception in thread main java.lang.AE: / by zero
        at Test.main()
```

Q.

Q1. Which three statements describe the object oriented featues of the java language?

A. Objects cannot be reused
B. A sub class can inherit from a super class
C. Objects can share behaviours with other objects
D. A package must contains more than one class
E. Object is the root class of all other objects
F. A main method must be declared in every class

Ans:B,C,E

Q

Q2. What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

A. Encapsulation
B. Inheritance
C. Abstraction
D. Instantiation
E. Polymorphism

Q3. Which statement best describes encapsulation?

A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects

B. Encapsulation enures that classes can be designed so that thier methods are inheritable

C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.

D. Encapsulation ensures that classes can be designed so that if a method has an argument X,any subclass of X can be passed to that methods.

res=>A

Q

Q4. Given the following two classes:

```
public class Customer
{
      ElectricAccount acct=new ElectricAccount();
      public void useElectricity(double kwh)
      {
            acct.addKwh(kwh);
      }
}
public class ElectricAccount
{
      private double kwh;
      public double rate=0.09;
      private double bill;
      //Line-1
}
```

How should you write methods in ElectricAccount class at Line-1 so that the member variable bill is always equal to the value of the member variable kwh multiplied by the member variable rate?

Any amount of electricity used by Customer(represented by an instance of the Customer class) must contribute to the Customer's bill(represented by member variable bill) through the method useElectricity() method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill?

A.

```
public void addKwh(double kwh)
{
        this.kwh+=kwh;
        this.bill=this.kwh*this.rate;
}
```

B.

```
public void addKwh(double kwh)
{
        if(kwh>0)
        {
                this.kwh+=kwh;
                this.bill=this.kwh*this.rate;
        }
}
```

C.

```
private void addKwh(double kwh)
{
        if(kwh>0)
        {
                this.kwh+=kwh;
                this.bill=this.kwh*this.rate;
        }
}
```

D.

```
public void addKwh(double kwh)
{
        if(kwh>0)
        {
                this.kwh+=kwh;
                setBill(this.kwh);
        }
}
public void setBill(double kwh)
{
        bill=kwh*rate;
}
```

ans=>B

Q.

Q5. Given the following code fragment:

```
public class Rectangle
{
        private double length;
        private double height;
        private double area;
        public void setLength(double length)
        {
                this.length=length;
        }
        public void setHeight(double height)
        {
                this.height=height;
        }
        public void setArea()
        {
                area=length*height;
        }
}
```

Which two changes would encapsulation this class and ensure that the area field is always equal to length*height,whenever Rectangle class is used?

A. Change the area field to public
B. Change the setArea() method to private?
C. Call the setArea() method at the beginning of the setLength() method
D. Call the setArea() method at the end of the setLength() method
E. Call the setArea() method at the beginning of the setHeight() method
F. Call the setArea() method at the end of the setHeight() method

res=>B,F

Q6. Given the following classes:

```
public class Employee
{
        public int salary;
}
public class Manager extends Employee
{
        public int budget;
}
public class Director extends Manager
{
        public int stockOptions;
}
```

And given the following main method:

```
public static void main(String[] args)
{
      Employee e = new Employee();
      Manager m = new Manager();
      Director d = new Director();
      //Line 1
}
```

Which two options fail to compile when placed at Line 1 of the main method?

A. e.salary=50_000;
B. d.salary=80_000;
C. e.budget=2_00_000;
D. m.budget=1_00_000;
E. m.stockOption=500;
F. d.stockOption=1_000;

ans=>C,E

Q.

```
abstract class Parent
{
    protected void resolve()//Line-1
    {
    }
    abstract void rotate();//Line-2
}
class Child extends Parent
{
    void resolve()//Line-3
    {
    }
    protected void rotate()//Line-4
    {
    }
}
```

Which two modifications,made independently, enable the code to compile?

A. Make that method at Line-1 public
B. Make that method at Line-2 public
C. Make that method at Line-3 public
D. Make that method at Line-3 protected
E. Make that method at Line-4 public

ans=>c d

Q.

Q8. Given:
Base.java:

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

DerivedA.java:

```
class DerivedA extends Base
{
        public void test()
        {
                System.out.println("DerivedA");
        }
}
```

DerivedB.java

```
class DerivedB extends DerivedA
{
	public void test()
	{
		System.out.println("DerivedB");
	}
	public static void main(String[] args)
	{
		Base b1= new DerivedB();
		Base b2= new DerivedA();
		Base b3= new DerivedB();
		b1=(Base)b3;
		Base b4=(DerivedA)b3;
		b1.test();
		b4.test();
	}

}
```

What is the result?

A.
Base
DerivedA

B.
Base
DerivedB

C.
DerivedB
DerivedB

ans=>c

Q.

Q9. Which two are benefits of polymorphism?

A. Faster Code at Runtime
B. More efficient Code at Runtime
C. More Dynamic Code at Runtime
D. More Flexible and Reusable Code at Runtime
E. Code that is protected from extension by other classes

ans=>c d

Q.

Q10. Which three statements are true about the structure of a Java class?

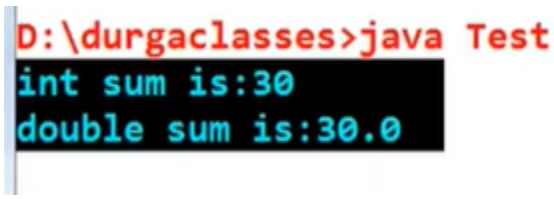
A) public class should compulsory contains main method
B) A class can have only one private constructor
C) A method can have the same name as variable
D) A class can have overloaded static methods
E) The methods are mandatory components of a class
F) The fields need not be initialized before use.

ans=>C F D

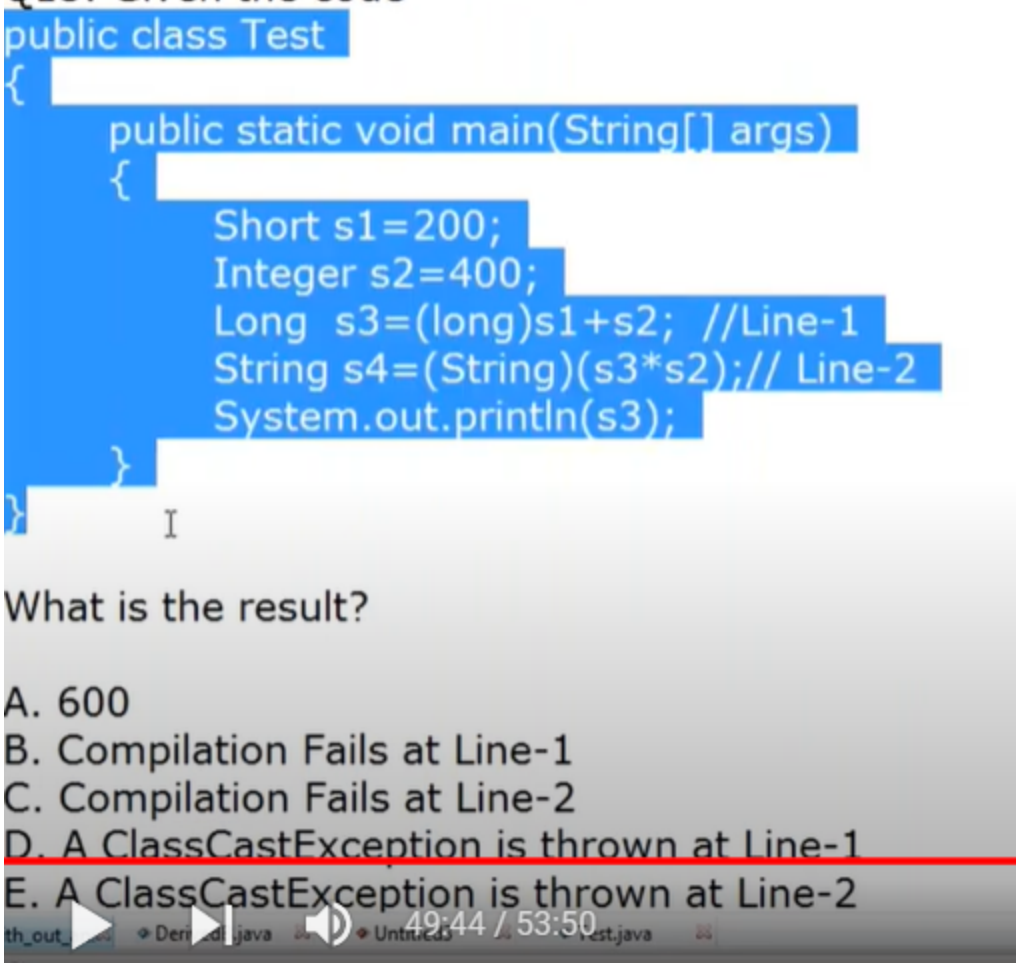
Q.

```java
public class Test
{
   public static void sum(Integer x,Integer y)
   {
      System.out.println("Integer sum is:"+(x+y));
   }
   public static void sum(double x,double y)
   {
      System.out.println("double sum is:"+(x+y));
   }
   public static void sum(float x,float y)
   {
      System.out.println("float sum is:"+(x+y));
   }
   public static void sum(int x,int y)
   {
      System.out.println("int sum is:"+(x+y));
   }
   public static void main(String[] args)
   {
      sum(10,20);
      sum(10.0,20.0);
```

ans=>



Q13. Given the code



ans=>c