

1. B, C, E
2. Encapsulation

OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 19-03-2018

11 8. Determine when casting is necessary  
12  
13 Q1. Which three statements describe the object oriented features of the java language?  
14  
15 A. Objects cannot be reused  
16 B. A sub class can inherit from a super class  
17 C. Objects can share behaviours with other objects  
18 D. A package must contains more than one class  
19 E. Object is the root class of all other objects  
20 F. A main method must be declared in every class  
21  
22  
23 Q2. What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?  
24  
25 A. Encapsulation  
26 B. Inheritance  
27 C. Abstraction  
28 D. Instantiation  
29 E. Polymorphism  
30  
31  
32 Q3. Which statement best describes encapsulation?

2:06 / 53:50

3. A

Option D is polymorphism

1 2 3 4 5 6 7 8 9 0

34 Q3. Which statement best describes encapsulation?  
35  
36 A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects  
37  
38 B. Encapsulation ensures that classes can be designed so that their methods are inheritable  
39  
40 C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.  
41  
42 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.

46

47 Q4. Given the following two classes:

48

49 public class Customer

50 {

51     ElectricAccount acct=new ElectricAccount();

52     public void useElectricity(double kwh)

53     {

54         acct.addKwh(kwh);

55     }

56 }

57 public class ElectricAccount

58 {

59     private double kwh;

60     public double rate=0.09;

61     private double bill;

62     //Line-1

63 }

65 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?

66

67 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?

We can decrease bill value by passing negative KWH . wrong

69 A.

70 public void addKwh(double kwh)

71 {

72     this.kwh+=kwh;

73     this.bill=this.kwh\*this.rate;

74 }

B is the right answer

75

76 B.

77 public void addKwh(double kwh)

78 {

79     if(kwh>0)

80     {

81         this.kwh+=kwh;

82         this.bill=this.kwh\*this.rate;

83     }

84 }

Private method cannot be accessed by electricity

85 C.

86 private void addKwh(double kwh)

87 {

88     if(kwh>0)

89     {

90         this.kwh+=kwh;

91         this.bill=this.kwh\*this.rate;

92     }

93 }

SetBill cannot be a method, has everything can be called by ElectricityMethod and this method don't call setBill

```

94 D.
95 public void addKwh(double kwh)
96 {
97     if(kwh>0)
98     {
99         this.kwh+=kwh;
100         setBill(this.kwh)
101     }
102 }
103 public void setBill(double kwh)
104 {
105     bill=kwh*rate;
106 }
107 I

```

Q5. Given the following code fragment:

```

84
85
86 public class Rectangle
87 {
88     private double length;
89     private double height;
90     private double area;
91     public void setLength(double length)
92     {
93         this.length=length;
94     }
95     public void setHeight(double height)
96     {
97         this.height=height;
98     }
99     public void setArea()
100     {
101         area=length*height;
102     }
103 }
104
105 Which two changes would encapsulation this class and ensure that the area field is always equal to
length*height, whenever Rectangle class is used?

```

Becarefull....

If you use D and F , u are calling SetArea twice. What u can do is change Set Area to private and call set area at the end of SetHeight method .. B and F

```

107 A. Change the area field to public
108 B. Change the setArea() method to private?
109 C. Call the setArea() method at the beginning of the setLength() method
110 D. Call the setArea() method at the end of the setLength() method
111 E. Call the setArea() method at the beginning of the setHeight() method
112 F. Call the setArea() method at the end of the setHeight() method
113

```

```
113 F. Call the setArea() method at the end of the setHeight() method
114
115 Q6. Given the following classes:
116 public class Employee
117 {
118     public int salary;
119 }
120 public class Manager extends Employee
121 {
122     public int budget;
123 }
124 public class Director extends Manager
125 {
126     public int stockOptions;
127 }
128
129 And given the following main method:
```

C and E

```
131 public static void main(String[] args)
132 {
133     Employee e = new Employee();
134     Manager m = new Manager();
135     Director d = new Director();
136     //Line 1
137 }
138 Which two options fail to compile when placed at Line 1 of the main method?
139
140 A. e.salary=50_000;
141 B. d.salary=80_000;
142 C. e.budget=2_00_000;
143 D. m.budget=1_00_000;
144 E. m.stockOption=500;
145 F. d.stockOption=1_000;
146
147 Q7. Given the code fragment:
```

```

1  abstract class Parent
2  {
3      protected void resolve();//Line-1
4      {
5      }
6      abstract void rotate();//Line-2
7  }
8  class Child extends Parent
9  {
10     void resolve();//Line-3
11     {
12     }
13     protected void rotate();//Line-4
14     {
15     }
16 }

```

Resolve – Protected to Default Not Possible – Make Line 3 Protected

Default -> Protected is possible – so no problem with Rotate Method

C and D. Sp Make Resolve either protected or Public'

Also if you change Line 4 to Public no harm, that is not an issue but its not throwing CE. This could be an allowed change but there is no issue with the existing

See which two modification made independently is -> if you made 1 change it should work and if you make the other change also it should work . Not both has to be made to make it work . Be carefull

```

17 Which two modifications,made indepeñdently, enable the code to compile?
18
19 A. Make that method at Line-1 public
20 B. Make that method at Line-2 public
21 C. Make that method at Line-3 public
22 D. Make that method at Line-3 protected
23 E. Make that method at Line-4 public

```

```

174 Q8. Given:
175 Base.java:
176
177 class Base
178 {
179     public void test()
180     {
181         System.out.println("Base");
182     }
183 }

```

```

185 DerivedA.java:
186
187 class DerivedA extends Base
188 {
189     public void test()
190     {
191         System.out.println("DerivedA");
192     }
193 }

```

```

194
195 DerivedB.java
196
197 class DerivedB extends DerivedA
198 {
199     public void test()
200     {
201         System.out.println("DerivedB");
202     }
203     public static void main(String[] args)
204     {
205         Base b1= new DerivedB();
206         Base b2= new DerivedA();
207         Base b3= new DerivedB();
208         b1=(Base)b3;
209         Base b4=(DerivedA)b3;
210         b1.test();
211         b4.test();
212     }
213 }

```

Here Line208= b1-> b3

B4-> B3. Both points to same object.

```

197 class DerivedB extends DerivedA
198 {
199     public void test()
200     {
201         System.out.println("DerivedB");
202     }
203     public static void main(String[] args)
204     {
205         Base b1= new DerivedB();
206         Base b2= new DerivedA();
207         Base b3= new DerivedB();
208         b1=(Base)b3;
209         Base b4=(DerivedA)b3;
210         b1.test();
211         b4.test();
212     }
213 }

```

What is the result?

A.

Base

Questions

Question	Asker	R...
sir line 4 we can make it public so why wrong ok get it	Shubham Gupta	
overriding	Shubham Gupta	
overriding	Pooja Chavan	



Same method having multiple implementation. Overriding

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 19-03-2018
1 polymorphism
2 -----
3 overloading
4 overriding
5
6 Flexibility
7 Dynamic Code
8
9
10 Reusability ==> Inheritance
11 Security ==> Encapsulation
```

```
236
237 Q9. Which two are benefits of polymorphism?
238
239 A. Faster Code at Runtime
240 B. More efficient Code at Runtime
241 C. More Dynamic Code at Runtime
242 D. More Flexible and Reusable Code at Runtime
243 E. Code that is protected from extension by other classes
```

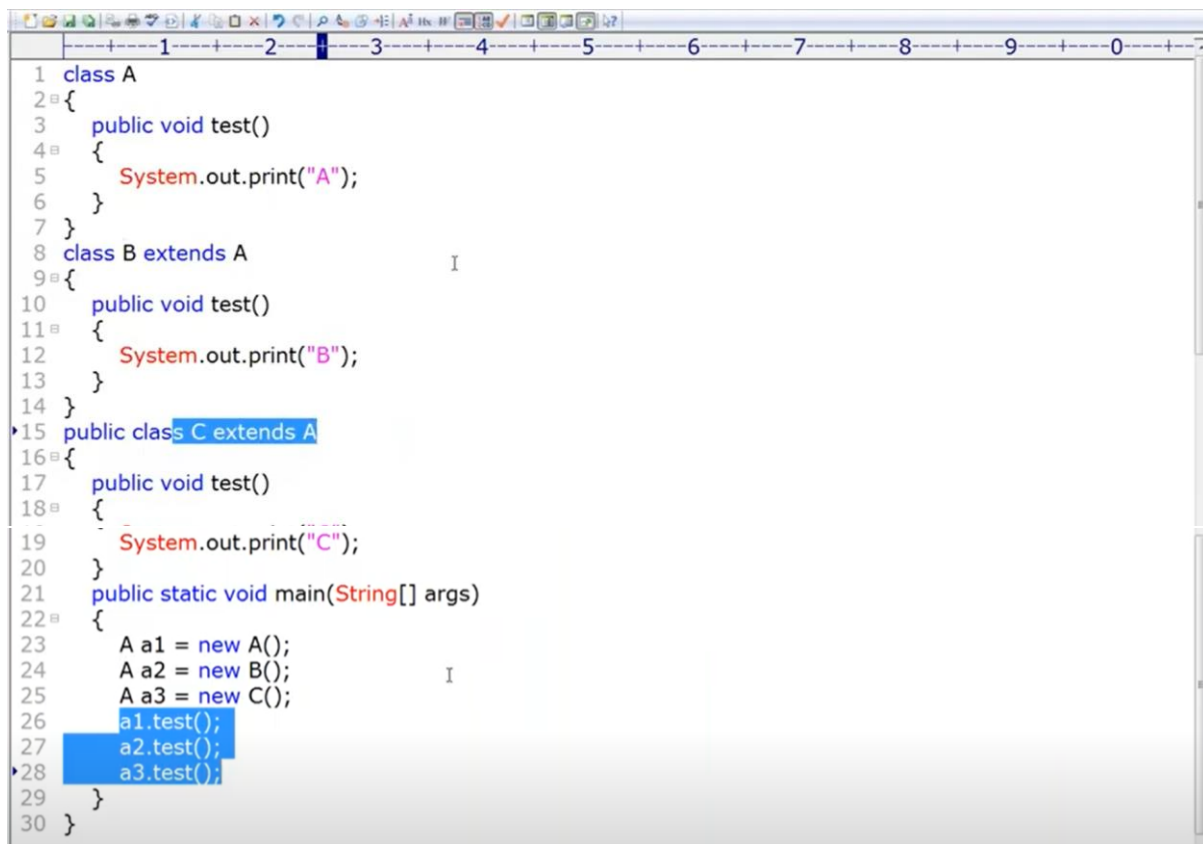
Variable and method can have the same name

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 19-03-2018
1 class Test
2 {
3     static int x1=10;
4     public static void x1()
5     {
6         System.out.println("x1 method ");
7     }
8     public static void main(String[] args)
9     {
10        System.out.println(x1);
11        x1();
12    }
13 }
14
```

C,D and F correct

```
246 Q10. Which three statements are true about the structure of a Java class?
247
248 A) public class should compulsory contains main method
249 B) A class can have only one private constructor
250 C) A method can have the same name as variable
251 D) A class can have overloaded static methods
252 E) The methods are mandatory components of a class
253 F) The fields need not be initialized before use.
```

A B C



```
1 class A
2 {
3     public void test()
4     {
5         System.out.print("A");
6     }
7 }
8 class B extends A
9 {
10    public void test()
11    {
12        System.out.print("B");
13    }
14 }
15 public class C extends A
16 {
17     public void test()
18     {
19         System.out.print("C");
20     }
21     public static void main(String[] args)
22     {
23         A a1 = new A();
24         A a2 = new B();
25         A a3 = new C();
26         a1.test();
27         a2.test();
28         a3.test();
29     }
30 }
```

If there is a public class in a file , then the file name should be the public class name

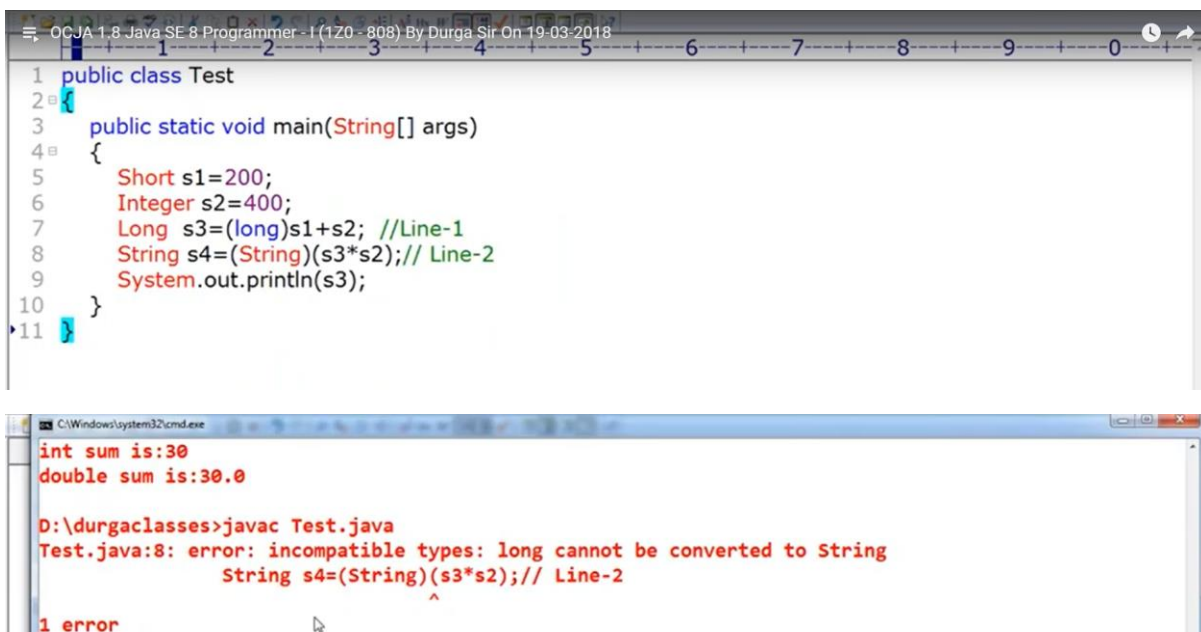


30 and 30.0

```
1 public class Test
2 {
3     public static void sum(Integer x,Integer y)
4     {
5         System.out.println("Integer sum is:"+(x+y));
6     }
7     public static void sum(double x,double y)
8     {
9         System.out.println("double sum is:"+(x+y));
10    }
11    public static void sum(float x,float y)
12    {
13        System.out.println("float sum is:"+(x+y));
14    }
15    public static void sum(int x,int y)
16    {
17        System.out.println("int sum is:"+(x+y));
18    }
19    public static void main(String[] args)
20    {
21        sum(10,20);
22        sum(10.0,20.0);
```

Line 1 no issues

Number cannot be typecasted to String



The screenshot shows an IDE window titled "OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 19-03-2018". The code in the editor is as follows:

```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         Short s1=200;
6         Integer s2=400;
7         Long s3=(long)s1+s2; //Line-1
8         String s4=(String)(s3*s2); // Line-2
9         System.out.println(s3);
10    }
11 }
```

Below the code editor, a command prompt window shows the output of the compilation and execution:

```
int sum is:30
double sum is:30.0

D:\durgaclass>javac Test.java
Test.java:8: error: incompatible types: long cannot be converted to String
        String s4=(String)(s3*s2); // Line-2
                        ^
1 error
```

