Started on	Wednesday, 1 April 2020, 4:19 PM
State	Finished
Completed on	Wednesday, 1 April 2020, 4:21 PM
Time taken	2 mins 22 secs
Marks	8.00/10.00
Grade	**80.00** out of 100.00
Feedback	Congratulations!! You have passed by securing more than 80%

Question 1

Correct

Mark 1.00 out of 1.00

What will be the output of the following program ?

```
class A
{
   public void test()
   {
      System.out.println("Class A");
   }
}
public class Trial extends A
{
   public void test()
   {
      System.out.println("Class Trial");
   }
   public static void main(String args[])
   {
      Trial object = (Trial)new A();
      object.test();
   }
}
```

Select one:

● a. Runtime Error ✓

○ b. Compile Time Error

○ c. Class Trial

○ d. Class A

Trial object = (Trial)new A(); produces run time exception because a parent object can never be referred by a child type reference.

The correct answer is: Runtime Error

Correct

Mark 1.00 out of 1.00

Observe the following code

```
class FourWheeler
{
  public void display()
  {
System.out.println("FourWheelers displayed");
  }
  public void get()
 {
 System.out.println("Get FourWheelers");
 display();
 }
}
class Car extends FourWheeler
{
  public void display()
 {
 System.out.println("Cars displayed");
 super.display();
 }

 public static void main(String [] args)
 {
FourWheeler f=new Car();
f.get();
 }

}
```

What is the output of the above code ?

Select one:

● a.
Get FourWheelers
Cars displayed
FourWheelers displayed ✓

○ b.
Get FourWheelers
FourWheelers displayed

○ c.
FourWheelers displayed
Cars displayed

○ d.
Get FourWheelers
FourWheelers displayed
Cars displayed

The main method has child object referred by parent reference. In this case, on invoking any method, only when a method is unavailable in the child, it is searched for in the parent.

The functions get executed in the following order.
get() of FourWheelers class
display() of Car class
diaplay() of FourWheelers class

The correct answer is:
Get FourWheelers
Cars displayed
FourWheelers displayed

Correct

Mark 1.00 out of 1.00

```
Class Icecream{
   public void displayName(String...s){
      System.out.println(s+” “ +“Icecream");
   }
   public void describe(String s) {
      System.out.println(s+” “ +"Icecream: Ice cream is a sweetened frozen food typically eaten as a snack or dessert. ");
   }
}
class Faloodeh extends Icecream {
   public void displayName (String s){
      System.out.println(s+” “ +"Faloodeh ");
   }

   public void describe (String s) {
      System.out.println(s+” “ +"Faloodeh:  Faloodeh is often served alongside Persian-style dairy-based ice cream ");
   }
}
public class Test {
   public static void main(String arg[]) {
     Icecream a=new Faloodeh ();
     Faloodeh b=( Faloodeh)a;
      a.displayName ("test"); b.displayName ("test");
      a. describe ("test");b. describe ("test");
   }
}
```

Select one:

○ a. test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh
test Faloodeh
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream

● b. test Faloodeh
test Faloodeh
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream ✓

○ c. test Faloodeh
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh

○ d. test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh
test Faloodeh

The main method has child object(Faloodeh) referred by parent reference(Icecream).

Then the parent reference is downcasted to child type reference and assigned to the new child reference.

On invoking displayName() with downcasted parent and child reference, the definition in child is implemented.

On invoking describe() with downcasted parent and child reference, the definition in child is implemented.

The correct answer is: test Faloodeh
test Faloodeh
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream
test Faloodeh: Faloodeh is often served alongside Persian-style dairy-based ice cream

Incorrect

Mark 0.00 out of 1.00

Select one:

● a. Exception occurs ✗

○ b. Compilation Error : we can't inherit the class which is “final”.

○ c. not necessary to create reference, automatically calls by the sub class reference.

○ d. Test t=new Test();

Your answer is incorrect.

The correct answer is: Compilation Error : we can't inherit the class which is “final”.

Question 5

Correct

Mark 1.00 out of 1.00

What will be the output of the program?

```
class Tree { }
class Pine extends Tree { }
class Oak extends Tree { }
public class Forest1
{
    public static void main (String [] args)
    {
        Tree tree = new Pine();
        if( tree instanceof Pine )
            System.out.println ("Pine");
        else if( tree instanceof Tree )
            System.out.println ("Tree");
        else if( tree instanceof Oak )
            System.out.println ( "Oak" );
        else
            System.out.println ("Oops ");
    }
}
```

Select one:

○ a. Tree

● b. Pine ✓

○ c. Forest

○ d. Oops

Since Pine object is held by Tree type reference - tree, the first condition in the if construct evalutes to true and hence, "Pine".

The correct answer is: Pine

Question 6

Correct

Mark 1.00 out of 1.00

The equals() method takes the reference of Object as parameter. State true or false.

Select one:

● True ✓

○ False

The correct answer is 'True'.

Correct

Mark 1.00 out of 1.00

Your answer is correct.

The correct answer is:

[final methods] can't be overridden.

Question 8

Not answered

Marked out of 1.00

Which three statements are true?

Select one or more:

☐ a. A protected method in class X can be overridden by a subclass of A only if the subclass is in the same package as X.

☐ b. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.

☐ c. A non-static public final method in class X can be overridden in any subclass of X.

☐ d. A private static method can be called only within other static methods in class X.

☐ e. A method with the same signature as a private final method in class X can be implemented in a subclass of X.

☐ f. A final method in class X can be abstract if and only if X is abstract.

☐ g. A protected method in class X can be overridden by any subclass of X.

Your answer is incorrect.

The correct answers are: A protected method in class X can be overridden by any subclass of X., A public static method in class X can be called by a subclass of X without explicitly referencing the class X., A method with the same signature as a private final method in class X can be implemented in a subclass of X.

Correct

Mark 1.00 out of 1.00

```
{
Calculator()
{
System.out.println("Basic arithmetic operation ");
}
Calculator (int x)
{
this();
System.out.println(x + " " + "is the only operand supplied");
}
Calculator(int x, int y)
{
this(5);
System.out.println("Two operands supplied are multiplied and the resultant is "+ x * y);
}
public static void main(String args[])
{
new Calculator(8, 10);
}
}
```

Select one:

○ a. Compilation error

○ b. 5 is the only operand supplied
Two operands supplied are multiplied and the resultant is 80
Basic arithmetic operation

○ c. Basic arithmetic operation
Two operands supplied are multiplied and the resultant is 80
5 is the only operand supplied

● d. Basic arithmetic operation
5 is the only operand supplied
Two operands supplied are multiplied and the resultant is 80 ✓

The main method has the two parameterized constructor call. the first statement within this definition is one parameterized constructor call. And the first statement within this is no parameterized constructor call.

Hence the output : Basic arithmetic operation , 5 is the only operand supplied, Two operands supplied are multiplied and the resultant is 80.

The correct answer is: Basic arithmetic operation
5 is the only operand supplied
Two operands supplied are multiplied and the resultant is 80

Correct

Mark 1.00 out of 1.00

```
1. class Dog { }
2. class Beagle extends Dog { }
3.
4. class Kennel {
5. public static void main(String [] arfs) {
6. Beagle b1 = new Beagle();
7. Dog dog1 = new Dog();
8. Dog dog2 = b1;
9. Beagle b3 = (Beagle) dog2; ✓
10. }
11. }
```

Which, inserted at line 9, will compile?

Beagle b4 = dog2; is wrong because a child type reference can never old the parent object or parent reference.
Beagle b3 = (Beagle) dog2; is right because a downcasted parent reference can be assigned to a child reference.

The correct answer is:

Given:

```
1. class Dog { }
2. class Beagle extends Dog { }
3.
4. class Kennel {
5. public static void main(String [] arfs) {
6. Beagle b1 = new Beagle();
7. Dog dog1 = new Dog();
8. Dog dog2 = b1;
9. [Beagle b3 = (Beagle) dog2;]
10. }
11. }
```

Which, inserted at line 9, will compile?