Question - 1
Java Keywords

Which keyword is used to take control out of the enclosing loop?

- for
- break
- if
- continue

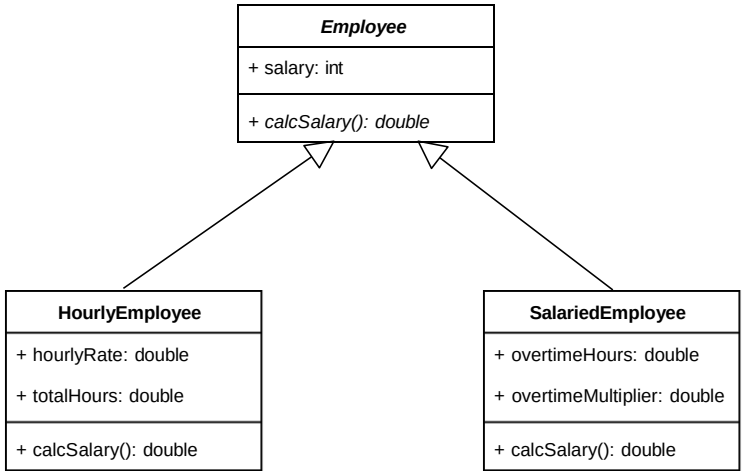
Question - 2
Java Constants

How is a constant declared so that it can be accessed without making an instance of a class in which it is declared?

- public final int intConst
- public static int intConst
- public static final int intConst
- public int intConst

Question - 3
Classes and Objects

Observe the following class diagram and state



Which of the following statements are true regarding it?

- Employee class is abstract and cannot be instantiated.
- Both HourlyEmployee and SalariedEmployee class should override the calcSalary() method, in order to get instantiated.
- Employee class cannot contains concrete methods.
- There is a composition relationship between Employee and the other two classes.

Question - 4
Java Interface

What is the output of the following program?

```
@FunctionalInterface
public interface MyInterface {
        void foo();
}
```

```
public class MyClass implements MyInterface {
    public void foo() {
        System.out.println("Hello");
    }
}
```

```
public class MyClass2 extends MyClass implements MyInterface{
    public void foo() {
        System.out.println("Hello 2");
    }
}
```

```
public class Main extends MyClass2{
    public static void main(String[] args) {
        MyClass c = new MyClass2();
        c.foo();
    }
}
```

- Hello 2
- Hello
- MyClass2 cannot extend a class and an interface at the same time.
- Runtime error

Question - 5
Java Operators

What is the value of the variable *z* in the statement `int x = 5; int y = x++; int z = ++y;` ?

- 5
- 6
- 7
- 8

Question - 6
Java Variables

What is the value of the variable *c* in the statement `int a = 5; int b = 6; int c = a = b;` ?

- 5
- 11
- 56
- 6

Question - 7
Java Inheritance Overriding

What will be the output of the following code block?

Animal.java

```
public abstract class Animal
{
        public abstract void makenoise();
}
```

Dog.java

```
public class Dog extends Animal
{
    public void makenoise()
    {
        System.out.println("Make Dog Noise...");
    }
}
```

GoldenRetriever.java

```
public class GoldenRetriever  extends Dog
{
    public void makenoise()
 {
        super.makenoise();
        System.out.println("Make Golden Retriever Noise");
 }
    public static void main(String ... arg)
 {
        Dog dog=new GoldenRetriever();
        dog.makenoise();
```

```
  }
}
```

- Make Dog Noise...
- Make Golden Retriever Noise
- Make Dog Noise...
  Make Golden Retriever Noise
- throws ClassCastException on runtime

Question - 8
Java String Pool

How many strings will be added to the string pool upon execution of the following code block?

```
public static void main(String[] args) {
        String str = "Hello There!";
        String mtr = "Hello" + " " + "There" + "!";
        String dtr = "Hello " + "There" + "!";
        String gtr = "Hello There" + "!";
    }
```

- 1
- 3
- 5
- 7
- 9

Question - 9
Which of the following operators is overloaded for string objects?

Which of the following operators is overloaded for string objects?

- +
- -
- ==
- >>
- Java doesn't support operator overloading

Question - 10

Select the correct statement, from the options below:

Select the correct statement, from the options below:

- If object obj1 can access object obj2 that is eligible for garbage collection, then obj1 is also eligible for garbage collection
- Objects can be explicitly destroyed using the keyword delete
- An object will be garbage collected immediately after it becomes unreachable
- None of the above

Question - 11

Given the following code, what is the most likely result:

Given the following code, what is the most likely result:

```
import java.util.*;
public class Compares
{
    public static void main( String args[])
    {
        String[] cities = {"Bangalore","Pune","San Francisco","New York City"};
        MySort ms = new MySort();
        Arrays.sort(cities, ms);
        System.out.println(Arrays.binarySearch(cities, "New York City"));
    }
    static class MySort implements Comparator
    {
        public int compare( String a, String b)
        {
            return b.compareTo(a);
        }
    }
}
```

- -1
- 1
- 2
- Compilation fails

Question - 12

Bitwise Operators in Java

Consider the following Java snippet:

int x = 3 & 5;
int y = 3 | 5;

The values of 'x' and 'y' (respectively) are:

- 1,7
- 2,8
- 8,-2
- 1,8
- 2,7

Question - 13
JDK Designe Patterns

One example of _____ Design pattern from JDK itself is the Collections.sort() method and the Comparator Interface, through which we can specify different criteria for sorting a collection of objects.

Which of the following is the best fit for the blank in this line?

- Singleton
- Adapter
- Factory
- Strategy

Question - 14
Value of k After Function Runs

Which is true of the following program?

```
public class TestFirstApp {
        static void doIt(int x, int y, int m) {
        if (x == 5) {
                m=y;
        } else {
                m=x;
        }
}

public static void main(String[] args) {
        int i=6, j=4, k=9;
        TestFirstApp.doIt(i,j,k);
        System.out.print(k);
}
```

- Doesn't matter what the values of i and j are, the output will always be 5 .
- Doesn't matter what the values of k and j are, the output will always be 5 .
- Doesn't matter what the values of i and j are, the output will always be 9 .
- Doesn't matter what the values of k and j are, the output will always be 9 .

Question - 15
Java Types

Which of the following is not a Java primitive type?

- String
- float
- double
- int
- byte

Question - 16
Method Overloading

```
public void foo(ArrayList<String> data)
{
    //some code
}
public void foo (ArrayList<Integer> data)
{
    //some code
}
public ArrayList<String> foo (ArrayList<String> data)
{
    //some code
}
private void foo(List<String> data)
{
    //some code
}
public void foo(ArrayList<String> data, boolean flag)
{
    //some code
}
```

Which of the statements are true regarding this code?

- All the methods are considered as overloaded methods, except for private void foo(List data) because it has a private access modifier.
- All methods can coexist without issue.
- 'foo(ArrayList)' clashes with 'foo(ArrayList)', as both methods have the same erasure.
- private void foo(List data) { } is an overloaded method, while public ArrayList foo (ArrayList data) { } is not.

Question - 17
Java BufferedReader

file1.txt

```
Hello

World
```

file2.txt

```
Hackerrank
```

What is the output of this code?

```
import  java.io.*;
public class Main{
    public static void main(String args[]) throws IOException {
        String str1, str2, str=null;
        File file1 = new File("file1.txt");
        File file2 = new File("file2.txt");
        BufferedReader br1 = new BufferedReader(new FileReader(file1));
        BufferedReader br2 = new BufferedReader(new FileReader(file2));
        while ((str1 = br1.readLine()) != null)
        {
            str2 = br2.readLine();
            if(str2 != null)
            {
                str=str1+" "+str2;
                System.out.println(str);
            }
            br1.close();
        }
    }
}
```

- Hello World
  IOException: Stream closed
- Hello Hackerrank
  World
- Hello World Hackerrank
- Hello Hackerrank
  IOException: Stream closed

Question - 18

Java parseInt

What is the output of the following code?

```
public class Main{
    public static void main(String[] args) {
        try
        {
            int[] a = new int[5];
            String s="1.0";
```

```
            try
            {
                System.out.println(Integer.parseInt(s+a[4]));
            }
            catch (Exception e)
            {
                System.out.println("Exception 1 occurs. "+ e.getClass());
            }
            throw new Exception();
        }
         catch (Exception e) {
            System.out.println("Exception 2 occurs");
        }
    }
}
```

- 1
  Exception 2 occurs
- Exception 1 occurs. class java.lang.NumberFormatException
  Exception 2 occurs
- Exception 1 occurs. class ArrayIndexOutOfBoundsException
  Exception 2 occurs
- Compile-Time Error

Question - 19

Java Design Pattern

The following code represents which design pattern?

```
class Student {
    private static Student s;
    private Student() {
    }
    public static Student getInstance() {
        if(s == null) {
            s = new Student();
        }
        return s;
    }
}
class Main {
    public static void main(String[] args) {
        Student s;
        s= Student.getInstance();
    }
}
```

- Singleton Design Pattern
- Factory Design Pattern
- Adapter Design Pattern
- Iterator Pattern

Question - 20

Java Concatenate

In main(), after creating an object of MyClass, which code option returns the string *s* concatenated with the sum of two numbers *x* and *y*?
For example, if *s = 'HackerRank', x = 2,* and *y = 3,* the output should be HackerRank 5

```
public interface MyInterface {
        String foo(String s, int x, int y);
    }
```

```
public class MyClass {
    public String foo2(MyInterface i, String s, int x, int y) {
        return i.foo(s, x, y);
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        MyClass myClass = new MyClass();
        // code to write
    }
}
```

- String result = myClass.foo2((s, x, y) -> s + " " + x + y, "HackerRank", 2, 3); System.out.println(result);
- String result = myClass.foo2((s, x, y) -> s + " " + (x + y), "HackerRank", 2, 3); System.out.println(result);
- String result = myClass.foo2->(s + " " + x + y, "HackerRank", 2, 3); System.out.println(result);
- String result = myClass.foo2((s, x, y) -> "HackerRank " + 2 + 3); System.out.println(result);

s and prints "Hello".