Question 1: Predict the output of following Java program?

```
class Main {
  public static void main(String args[]) {
    try {
      throw 10;
    }
    catch(int e) {
      System.out.println("Got the  Exception " + e);
    }
 }
}
```

Options

(A) Got the Exception 10

(B) Got the Exception 0

(C) Compiler Error

Answer: (C)

Explanation: In Java only throwable objects (Throwable objects are instances of any subclass of the Throwable class) can be thrown as exception. So basic data type cannot be thrown at all. Following are errors in the above program.

Question 2: what will be the output of given code?

```
class Test extends Exception {
}
 class Main {
  public static void main (String args[]) {
    try {
      throw new Test();
    }
    catch(Test t) {
      System.out.println("Got the Test Exception");
```

```
    }
    finally {
      System.out.println("Inside finally block ");
    }
  }
}
```

Options

(A) Got the Test Exception
Inside finally block

(B) Got the Test Exception

(C) Inside finally block

(D) Compiler Error

Answer: (A)

Explanation: In Java, the finally is always executed after the try-catch block. This block can be used to do the common cleanup work. There is no such block in C++.

Question 3: Output of following Java program?

```
class Main
 { public static void main(String args[]) {
    int x = 0;
    int y = 10;
    int z = y/x;
  }
}
```

Options

(A) Compiler Error

(B) Compiles and runs fine

(C) Compiles fine but throws ArithmeticException exception

Answer: (C)

Explanation: ArithmeticException is an unchecked exception, i.e., not checked by the compiler. So the program compiles fine. See following for more details.

Question 4: what will be the output of given code?

```
class Test
{ public static void main (String [] args)
   {
      try
      {
         int a = 0;
         System.out.println ("a = " + a);
         int b = 20 / a;
         System.out.println ("b = " + b);
      }
      catch(ArithmeticException e) {
         System.out.println ("Divide by zero error");
      }
      finally
      {
         System.out.println ("inside the finally block");
      }
   }
}
```

Options

(A) Compile error

(B) Divide by zero error

(C) a = 0

Divide by zero error
Inside the finally block
(D) a = 0
(E) Inside the finally block
Answer: (C)

Explanation: On division of 20 by 0, divide by zero exception occurs and control goes inside the catch block. Also, the finally block is always executed whether an exception occurs or not.

Question 5: what will be the output of given code ?

_____

```
class Test
{
   public static void main(String[] args)
   {
      try
      {
         int a[]= {1, 2, 3, 4};
         for (int i = 1; i <= 4; i++)
         {
            System.out.println ("a[" + i + "]=" + a[i] + "n");
         }
      }

      catch (Exception e)
      {
         System.out.println ("error = " + e);
      }

      catch (ArrayIndexOutOfBoundsException e)
```

```
    {
       System.out.println ("ArrayIndexOutOfBoundsException");
    }
  }
}
```

Options

(A) Compiler error
(B) Run time error
(C) ArrayIndexOutOfBoundsException
(D) Error Code is printed
(E) Array is printed

Answer: (A)

Explanation: ArrayIndexOutOfBoundsException has been already caught by base class Exception. When a subclass exception is mentioned after base class exception, then error occurs.