Data types
==========
In real world how many types of data we actually come cross?
 1. Number
      a. Integral type
            byte,short,int,long
      b. Floating type
            float,double
 2. Character
      a. char
 3. Boolean type
      a. boolean
 4. Images,audio,video
      a. LOB's(Large Object)

What is the need of dataype is java langauge?
 => Every variable has a type,every expression has a type and all types are strictly defined,moreover every assignement
    should be checked by the compiler for type compatiblity, hence java langauge is strictly typed programming language.
 => Since the language is strictly typed, we expect the programmer to specify the type of data we are keeping inside the
    variable.
 => To tell what type of data is kept inside the variable to the langauge, we need to tell through "DataType".

eg::
      int a;
        int b = a+10;

Is java a dynamic programming language?
 => Java is dynamic programming language,because memory for the datatype is allocated by the JVM at the runtime.

      int a = 10;
         |=> datatype(used by compiler to check for compatability)
       |=> During the execution of the program,jvm sees int, allocates 4 bytes of memory as per datatype specification
             and the value 10 will be stored inside 4bytes memory and for that 4bytes memory name is assigned as "a".

Dynamically typed language
==========================
 a = 10;
 a ="sachin";
 a = 54.5;
 a = true;
eg:: python,javascript,php,......

datatypes
=========
byte => size :: 1bytes
        compatable value(range) :: -128 to +127

eg::
byte a = 10;
byte b = 20;
byte c = a+b;(byte+byte)//CompileTimeError

```
System.out.println(c);


short => size   :: 2bytes
        range ::-32768 to +32767

int => size  :: 4bytes
       range :: -21474.... to +21474.....

long(out of scope)
  => size :: 8bytes
  => remember this datatype is normally used when we work with file operations

float => size :: 4bytes
      => It is capable of holding the data of precession upto 5decimal places.

double => size :: 8bytes
       => It is capable of holding the data of precession upto 14 decimal places.

typecasting
===========
    byte(1)----> short(2)----> int(4)-----> long(8) ------> float(4) -------> double(8)
                            ^
                            |
                          char(2)

long type of data how can it go and sit float datatype?
 => internal represenation of float datatype permits long datatype value to go and sit, so long values can accomdate in float
    type also.

ClassRoom
  8 Benches ---> each bench  -> 1 students ----> total students :: 8

ClassRoom
  4 Benches ---> each bench  -> 2 students ----> total students :: 8


boolean and char datatypes
==========================
  In java langauge to store the datatype of character how many bytes are required?
   => In java langauge,we give 2 types to store a character.

  Why 2 bytes to store one character in java?
  => java language support i18N,so we need 2 bytes to store a character.
  => i18N stands for "Internationalization".
  => If we build an application in java, the same application through i18N can be communicated across the globe as per there
     "LocalLanguage".
           eg:: Amazon ----> english,hindi,kannada,tamil,telgu,.......

1 byte -> 8 bits ---> -128 to +127(only these many characters it can support basically english language ,followed by few symobls)
2 byte -> 16 bits -->  more volume means more character support,.....



boolean datatypes
```

```
=================
  => The only allowed values for boolean datatype is "true and false".
  => What is size of booelan datatype?
     Ans. The boolean datatype size is machine depenent can't be predicted.
  => MachineDependent
       if we install jvm of windows machine,mac machine,linux machine based on the
architecutre of
         cpu, jvm will allocate the memory of that particular size for boolean type
variable.

eg::
public class Sample
{
      public static void main(String[] args)
      {
            boolean a  = 0;//CE
            boolean b  = 1;
            System.out.println(a+b);
      }
}

eg::
boolean b= 0;//CE
if (b)
      System.out.println("sachin");
else
      System.out.println("dhoni");


Snippets
========
Given code of Test.java file:
public class Test {
    public static void main(String[] args){
        args[1] = "Day!";
        System.out.println(args[0] + " " + args[1]);
    }
}
And the commands:
javac Test.java
java Test Good

What is the result?
A. Good
B. Good Day!
C. Compilation Error
D. JVM would create a problem during execution

Answer: D ( Array size can't be increase at the runtime)

8.
Consider below code of Test.java file:
public class Test {
    public static void main(String[] args){
        System.out.println("Welcome " + args[0] +"!");
    }
}
And the commands:
javac Test.java
```

```
java Test "James Gosling"  "Bill Joy"
```

What is the result?
A. Welcome James Gosling!
B. Welcome Bill Joy!
C. Welcome "James Gosling"!
D. Welcome "Bill Joy"!
E. Welcome James!
F. Welcome Gosling!
G. Welcome Bill!
H. Welcome Joy!

Answer: A

Consider below code of Test.java file:

```
public class Test {
    public static void main(String[] args) {
        boolean b1 = 0;
        boolean b2 = 1;
        System.out.println(b1 + b2);
    }
}
```

What is the result of compiling and executing Test class?
A. 0
B. 1
C. true
D. false
E. compilation error

Answer: E

10.
Given:

```
35. String #name = "Jane Doe";
36. int $age = 24;
37. Double _height = 123.5;
38. double ~temp = 37.5;
```

Which two statements are true? (Choose two.)
A. Line 35 will not compile.
B. Line 36 will not compile.
C. Line 37 will not compile.
D. Line 38 will not compile.

Answer: A and D

11.
What will be the result of compiling and executing Test class?

```
public class Test {
    public static void main(String[] args){
        byte b1 = ( byte ) ( 127 + 21 );
        System.out.println(b1);
    }
}
```

A. 148
B. Compilation Error
C. -108
D. -128

Answer :: C [minRange + (value-MaxRange) - 1]
[Reslut = -128 + 148 - 127-1]
= -108

12.

```
class Test
{
      public static void main(String[] args)
      {
            byte b[] = new byte[0];
            System.out.println(b[0]);

      }
}
```

A. NullPointerException
B. ArrayIndexOutOfBoundsException
C. Compilation Error
D. 0

Answer: B

13.

```
class Test
{
      public static void main(String[] args)
      {
            System.out.println("Hello" + 1 + 2 + 3 + 4);

      }
}
```

A. Hello1234
B. Hello9
C. Hello10
D. Hello 10

Answer: A

14.

```
class Test
{
      public static void main(String[] args)
      {
            System.out.println("Hello" + (1 + 2 + 3 + 4));

      }
}
```

A. Hello1234
B. Hello9
C. Hello10
D. Hello 10

Answer: C

15.

```
class Test
{
      public static void main(String[] args)
      {
```

```
            String msg = "hello";
            boolean flag[] =new boolean[1];//flag[0] = false
            if (flag[0])
            {
                  msg = "welcome";
            }
            System.out.println(msg);

      }
}
```

A. ArrayIndexOutOfBoundsException
B. Welcome
C. hello
D. NullPointerException

Answer: C

16.

```
class Test
{
      public static void main(String[] args)
      {
            String msg = "hello";
            Boolean flag[] =new Boolean[1];//flag[0] = null
            if (flag[0])
            {
                  msg = "welcome";
            }
            System.out.println(msg);

      }
}
```

A. ArrayIndexOutOfBoundsException
B. Welcome
C. hello
D. NullPointerException
E. CompileTimeError
F. None of the above

17.
Q>

```
class Fork {
      public static void main(String[] args) {
            if(args.length == 1 | args[1].equals("test")) {
                        System.out.println("test case");
            } else {
                  System.out.println("production " + args[0]);
            }
      }
}
```

And the command-line invocation:

```
java Fork live2
```

```
args[0] = "live2";
args.length = 1
```

What is the result?

A. test case
B. production live2
C. test case live2
D. Compilation fails
E. An exception is thrown at runtime

Answer:: E(args[1] :: doesnot exists so problem at the runtime)

What is the difference b/w "||" and "|"?
  "||" => short circuit operator
   "|" => Bitwise operator

Tommo questions will be on :: Operator, control statements and oops concepts