**1) A command-line argument in Java is a value passed at the time of \_\_\_ a program.**

A) Compiling

B) Running

C) -

D) -

Answer [=]

**B**

**Explanation:**

**Yes, Command-line arguments are passed at the time of running.**

**2) Command-line arguments help in a way of \_\_\_\_ data to a program.**

A) Inputting

B) Outputting

C) -

D) -

Answer [=]

**A**

**Explanation:**

**Inputting is nothing but supplying data to a program.**

**3) The command-line arguments are passes at \_\_\_\_.**

A) Runtime

B) the time of executing a Java program.

C) the time of compiling a Java program.

D) None

Answer [=]

**B**

**Explanation:**

**"Runtime" is different from the "time of running a program". Runtime is dynamic.**

**4) When a Java program is executed once, how many times can you pass data using the Command-line arguments?**

A) 1

B) 8

C) 16

D) None

Answer [=]

**A**

**Explanation:**

**Only once.**

**5) If you need to accept data at runtime, you use \_\_\_ in Java.**

A) Command-line arguments

B) Java IO statements

C) -

D) -

Answer [=]

**B**

**Explanation:**

**Java IO statements are Input-Output statements which are part of java.io.\* package.**

**6) Which is the method that accepts data passed in the form of command-line arguments in Java?**

A) show() method

B) main() method

C) display() method

D) print() method

Answer [=]

**B**

**Explanation:**

**Command-line arguments are useful only when you are using and depending on a MAIN() method.**

**7) The command-line arguments in Java are used along with a \_\_\_\_ command.**

A) javac

B) java or javaw

C) javap

D) All the above

Answer [=]

**B**

**Explanation:**

**You can actually pass string type data with java or javaw command only.**

**8) The type of Arguments the MAIN method accepts is \_\_\_.**

A) Integer[]

B) Float[]

C) Long[]

D) String[]

Answer [=]

**D**

**Explanation:**

**Yes. The main method accepts String array data.**

**9) The data that is passed at the time of running a Java program as command-line arguments are converted into \_\_\_ data type.**

A) Integer array

B) Float array

C) Character array

D) String array

Answer [=]

**D**

**Explanation:**

**Any type of data that can be typed in command prompt as command-line arguments are converted finally into a String array.**

**10) The delimiter used to separate command-line arguments in Java is \_\_\_\_.**

A) Semicolon (;)

B) Colon (:)

C) Space

D) Comman (,)

Answer [=]

**C**

**11) Can you pass a sentence with multiple words separated by spaces as a single command-line argument in Java?**

A) Yes

B) No

C) -

D) -

Answer [=]

**A**

**Explanation:**

**You should keep the string or sentence within double quotes to tell the JVM that you are passing a single command-line argument.**

**12) What is the output of the below Java program with command-line arguments?**

**public class CommandLineArgs1**

**{**

**public static void main(String[] args)**

**{**

**for(String str: args)**

**{**

**System.out.println(str);**

**}**

**if(args.length == 0)**

**System.out.println("No arguments passed");**

**}**

**c:\folder>java CommandLineArgs1 car brake horn**

A)

car brake horn

B)

car

brake

horn

C)

horn brake car

D)

horn

brake

car

Answer [=]

**B**

**Explanation:**

**Each space tells the JVM to accept a new argument. So, in the above example, we have passed 3 arguments.**

**13) Is there any limit to the number of spaces between two arguments of command-line arguments in Java?**

A) Yes

B) No

C) -

D) -

Answer [=]

**B**

**Explanation:**

**You can keep any number of spaces between two adjacent arguments.**

**14) To pass a string as a command-line argument in Java, you need to surround the text within a pair of \_\_\_.**

A) Single Quotes ('abc def')

B) Double Quotes ("abc def")

C) Double Spaces(  abc def  )

D) Triple Single Quotes ('''abc  def''')

Answer [=]

**B**

**Explanation:**

**Double quotes tell the JVM to interpret all the text with words as a single string.**

**15) Choose the correct way of receiving command-line arguments with in the MAIN method in Java?**

A)

public static void main(String[] args)

{

}

B)

public static void main(String args[])

{

}

C)

public static void main(String anyName[])

{

}

D) All the above.

Answer [=]

**D**

**Explanation:**

**You can give any name to the String-array variable accepting command-line arguments.**

**16) what is the output of the below Java program with command-line arguments?**

**public class CommandLineArguments2**

**{**

**public static void main(String[] args)**

**{**

**System.out.println(args[1]);**

**}**

**}**

**C:\folder>java CommandLineArguments2 TIGER**

A) CommandLineArguments2

B) TIGER

C) Compiler error

D) None

Answer [=]

**C**

**Explanation:**

**Program name or class-name is not considered as a command-line argument. In the above example, one argument is passed (TIGER) which can be accessed by args[0].**

**17) Which is the exception or error that is thrown if a non-existing command-line argument is referred to in a Java program?**

A) StackOverflowError

B) IndexOutOfBoundsException

C) ArrayIndexOutOfBoundsException

D) ArithmeticException

Answer [=]

**C**

**Explanation:**

**As the command-line arguments are processed as a String-array, the exception will be ArrayIndexOutOfBoundException only.**

**18) Any type of data that can be typed on a console or ECLIPSE can be passed as a command-line argument. State TRUE or FALSE.**

A) TRUE

B) FALSE

C) -

D) -

Answer [=]

**A**

**Explanation:**

**TRUE**

**19) Which are the methods used to parse string values to the respective data types in Java?**

A) Boolean.parseBoolean(), Byte.parseByte()

B) Integer.parseInt(), Long.parseLong()

C) Float.parseFloat(), Double.parseDouble()

D) All the above

Answer [=]

**D**

**Explanation:**

**All the above are static methods used to parse String values to the respective types. Parsing is nothing but converting.**

**20) What is the output of the below Java program?**

**public class CommandLineArguments3**

**{**

**public static void main(String[] args)**

**{**

**String name = args[0];**

**int age = Integer.parseInt(args[1]);**

**Boolean married = Boolean.parseBoolean(args[2]);**

**Float salary = Float.parseFloat(args[3]);**

**System.out.println("Name="+name + ", Age=" + age + ", Married=" + married + ", Salary=$"+ salary);**

**}**

**}**

**C:\folder>java CommandLineArguments3 Marker 25 false 5025.35**

A) Name=Marker, Age=25, Married=false, Salary=$5025.35

B) Name=Marker, Age=25, Married=false, Salary=$5025

C) Compiler error

D) None

Answer [=]

**A**

**Explanation:**

**The order of parsing is very important. In the above program, we carefully chose the argument index and parsed correctly.**