1. What is the primary function of the JVM?

A) Compile Java code
B) Execute Java bytecode
C) Write Java code
D) Debug Java code
Answer: B) Execute Java bytecode

2. Which of the following is a feature of Java?

A) Platform dependent
B) Supports only object-oriented programming
C) Automatic memory management
D) Requires manual memory management
Answer: C) Automatic memory management

3. What type of variable is declared inside a method?

A) Instance variable
B) Static variable
C) Local variable
D) Global variable
Answer: C) Local variable

4. Which Java data type is used to store a true/false value?

A) int
B) double
C) boolean
D) char
Answer: C) boolean

5. Which of the following is NOT a control statement in Java?

A) if
B) while
C) for
D) string
Answer: D) string

---

1. Which of the following is the correct syntax to create a class in Java?

A) `class MyClass { }`
B) `create class MyClass { }`
C) `class: MyClass { }`
D) `MyClass class { }`
Answer: A) `class MyClass { }`

2. Which of the following is true about a local variable in Java?

A) Local variables are accessible globally.
B) Local variables are initialized automatically.
C) Local variables must be explicitly initialized before use.
D) Local variables can be accessed outside the method.
Answer: C) Local variables must be explicitly initialized before use.

3. Which of the following is NOT a type of operator in Java?

A) Arithmetic
B) Assignment
C) Comparison
D) Execution
Answer: D) Execution

4. What does the 'break' statement do in a loop?

A) Continues with the next iteration of the loop
B) Exits the loop
C) Skips the current iteration
D) Pauses the loop
Answer: B) Exits the loop

5. Which of the following is the correct way to declare a constant in Java?

A) `int constantValue = 100;`
B) `const int constantValue = 100;`
C) `final int constantValue = 100;`
D) `final constant int constantValue = 100;`
Answer: C) `final int constantValue = 100;`

Difficult Level

1. Which of the following statements is true about wrapper classes in Java?

A) Wrapper classes are immutable.
B) Wrapper classes are used to convert primitive types to objects.
C) Wrapper classes cannot be used with generics.
D) Wrapper classes cannot be used with methods.
Answer: B) Wrapper classes are used to convert primitive types to objects.

2. Which of the following statements is correct about JVM architecture?

A) The JVM includes the JDK and the JRE.
B) The JVM is platform-dependent.
C) The JVM directly interacts with hardware.
D) The JVM uses source code directly for execution.
Answer: A) The JVM includes the JDK and the JRE.

4. What is the purpose of the 'instanceof' operator in Java?

A) To check if an object is null
B) To compare the values of two objects
C) To check if an object is an instance of a particular class or subclass
D) To compare primitive types
Answer: C) To check if an object is an instance of a particular class or subclass

1. Which of the following is used to organize classes into packages in Java?

A) import
B) package
C) class
D) module
Answer: B) package

2. How do you declare an array in Java?

A) `int arr[];`
B) `int[] arr;`
C) Both A and B
D) None of the above
Answer: C) Both A and B

3. Which of the following methods is used to add a string to a `StringBuilder` object?

A) `add()`
B) `append()`
C) `concatenate()`
D) `insert()`
Answer: B) `append()`

4. Which of the following access modifiers allows access only within the same package?

A) `public`
B) `private`
C) `protected`
D) `default`
Answer: D) default

5. Which of the following is used to define a constructor in Java?

A) `constructor MyClass() {}`
B) `MyClass() {}`
C) `void MyClass() {}`
D) `class MyClass() {}`
Answer: B) `MyClass() {}`

Difficult Level

1. Which of the following is the correct way to access a class in a package `com.example`?

A) `import com.example.*;`
B) `package com.example.*;`
C) `import com.example;`
D) `import * from com.example;`
Answer: A) `import com.example.*;`

2 What will be the output of the following code?

```
String str = "hello";
str = str.concat(" world");
System.out.println(str);
```

A) `hello`
B) `hello world`
C) `world`
D) Compile-time error
Answer: B) `hello world`

3. What is the primary benefit of using a `StringBuilder` in Java?

A) StringBuilder objects are immutable
B) StringBuilder objects are mutable and efficient for string concatenation
C) StringBuilder is faster for string comparison
D) StringBuilder cannot store null values
Answer: B) StringBuilder objects are mutable and efficient for string concatenation

4

```
class Person {
  private String name;
  private int age;

  public Person(String name, int age) {
    this.name = name;
    this.age = age;
  }

  public String getName() {
    return name;
  }

  public int getAge() {
    return age;
```

```
    }
}
```
What is the concept demonstrated by this code?

A) Polymorphism
B) Inheritance
C) Encapsulation
D) Method Overloading
Answer: C) Encapsulation

Which of the following statements about method overloading in Java is true?

A) Methods must have different return types to be overloaded.
B) Methods can be overloaded by changing the order of parameters.
C) Method overloading occurs only when methods have the same name but different access modifiers.
D) Method overloading is not supported in Java.
Answer: B) Methods can be overloaded by changing the order of parameters.

6. What is the key difference between a `String` and a `StringBuilder` in Java?

A) `StringBuilder` is immutable, while `String` is mutable.
B) `String` is mutable, while `StringBuilder` is immutable.
C) `String` is immutable, while `StringBuilder` is mutable.
D) Both `String` and `StringBuilder` are mutable.
Answer: C) `String` is immutable, while `StringBuilder` is mutable.

7. Which of the following is true about immutable classes in Java?

A) They can have mutable fields.
B) Their state cannot be changed once created.
C) They must have at least one constructor.
D) They allow modification of their fields directly.
Answer: B) Their state cannot be changed once created.

8 what is the output?

```
// filename Test.java
class Test {
  public static void main(String args[]) {
     System.out.println(fun());
  }
  static int fun() {
     static int x= 0;
```

```
      return ++x;
   }
}
```

A Compile time error

B Run time error

C 0

D 1

Answer A

Q1. What will be the output of the program?

```
public class ObjComp
{
   public static void main(String [] args )
    {
       int result = 0;
       ObjComp oc = new ObjComp();
       Object o = oc;

       if (o == oc)
          result = 1;
       if (o != oc)
          result = result + 10;
       if (o.equals(oc) )
          result = result + 100;
       if (oc.equals(o) )
          result = result + 1000;

       System.out.println("result = " + result);
    }
```

}

1. 1
2. 10
3. 101
4. 1101

Q2. What will be the output of the program?

int i = (int) Math.random();

1. i = 0
2. i = 1
3. value of i is undetermined
4. Statement causes a compile error

Q3. Which one of these lists contains only Java programming language keywords?

(a) class, if, void, long, Int, continue

(b) goto, instanceof, native, finally, default, throws

(c) try, virtual, throw, final, volatile, transient

(d) strictfp, constant, super, implements, do

Q4. What will be the output of the program?

```
class Test
{
   public static void main(String [] args)
   {
      int x= 0;
      int y= 0;
      for (int z = 0; z < 5; z++)
      {
         if (( ++x > 2 ) && (++y > 2))
         {
            x++;
```

```
        }
      }
      System.out.println(x + " " + y);
   }
}
```

1. 5 2
2. 5 3
3. 6 3
4. 6 4

Q5. Given the following piece of code:

```
class Person{
public void talk(){}
 }
public class Test{
 public static void main(String args[])
{
Person p = null;
 Try
{
 p.talk();
}
catch(NullPointerException e){ System.out.print("There is a NullPointerException. ");}
catch(Exception e){  System.out.print("There is an Exception. ");   }
 System.out.print("Everything went fine. ");
 }
}
```

what will be the result?

(a) If you run this program, the outcome is:There is a NullPointerException. Everything went fine.

(b) If you run this program, the outcome is: There is a NullPointerException.

(c) If you run this program, the outcome is:There is a NullPointerException. There is an Exception.

(d) This code will not compile, because in Java there are no pointers.

Q6. What is the result of this program?

```
class Over{
public static void main(String[] args){
Under u = new Under();
u.test();}
int test(){
System.out.println("over");
return 1;}}
class Under extends Over{
short test(){
super.test();
System.out.println("Under");
return 1;}}
```

1. This code compiles, runs and displays over followed by Under
2. This code compiles, runs and displays Under followed by over
3. This code does not compile
4. Code will compile but gives runtime error

Q7. Which is valid declaration within an interface?

1. public static short stop = 23
2. protected short stop = 23
3. transient short stop = 23;
4. final void madness(short stop);

Q8. Which collection class allows you to access its elements by associating a key with an element's value, and provides synchronization?

1. java.util.SortedMap
2. java.util.TreeMap
3. java.util.TreeSet
4. java.util.HashTable

Q9. class X implements Runnable {

```
public static void main(String args[])
{
   /* Missing code? */
}
public void run() {} }
```

Which of the following line of code is suitable to start a thread ?

1. Thread t = new Thread(X);
2. Thread t = new Thread(X); t.start();
3. X run = new X(); Thread t = new Thread(run); t.start();
4. Thread t = new Thread(); x.run();

Q10.Which method registers a thread in a thread scheduler?

1. run()
2. construct()
3. start()
4. register()

Q11. What will be output of following program?

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

1. 129
2. 2
3. -127
4. Compiler error

Q12. Which is not true java statement?

1. Java deallocates memory automatically.
2. Finalize method is just called before garbage collection.
3. Garbage collection runs when there is reference with object and runs periodically.
4. Inside finalize method we keep those code which must be executed before object is destroyed by garbage collection.

Q13. public class This {

```
    int i;

   This(){
       this.i++;
       i++;
   }

   public static void main(String[] args){
       This o=new This();
       System.out.print(o.i);
   }
}
```

What will be output of above program?

1. Garbage value
2. 1
3. 2
4. Non static variable must be initialized, Compiler error

Q14. What will be the output of the program?

```
try
{
   int x = 0;
   int y = 5 / x;
}
```

```
catch (Exception e)
{
   System.out.println("Exception");
}
catch (ArithmeticException ae)
{
   System.out.println(" Arithmetic Exception");
}
System.out.println("finished");
```

1. finished
2. Exception
3. Compilation fails
4. Arithmetic Exception

Q15. What allows the programmer to destroy an object x?

1. x.delete()
2. x.finalize()
3. Runtime.getRuntime().gc()
4. Only the garbage collection system can destroy an object.

Q16. Which of the following is true about ArrayList and Vector?

1. ArrayList is NOT synchronized by default where as Vector List is synchronized by default.
2. ArrayList can use only Iterator to access the elements whereas Vector list can use Iterator and Enumeration Interface to access the elements.
3. ArrayList can grow dynamically where as Vector list cannot grow dynamically.
4. ArrayList has no default size. Where as While vector has a default size of 1

Q17. What is a native method?

1. A native method is a method that is implemented in a language other than Java.
2. A native method is a method that is implemented in a language other than java but can be called from inside your java program.
2. A native method is a method that is defined in the inner class.

3. A native method is a method that is implemented by the subclass.

Q18. What will be the output of the program?

```
public class RTExcept
{
   public static void throwit ()
   {
      System.out.print("throwit ");
      throw new RuntimeException();
   }
   public static void main(String [] args)
   {
      try
      {
         System.out.print("hello ");
         throwit();
      }
      catch (Exception re )
      {
         System.out.print("caught ");
      }
      finally
      {
         System.out.print("finally ");
      }
      System.out.println("after ");
   }
}
```

1. hello throwit caught
2. Compilation fails
3. hello throwit RuntimeException caught after

4. hello throwit caught finally after

Q19. The use of protected keyword to a member in a class will restrict its visibility as follows:

1. Visible only in the class and its subclass in the same package.
2. Visible in all classes in the same package and subclasses in other packages.
3. Visible only inside the package.
4. Visible only in the class where it is declared.

1. What is the primary function of the Java Virtual Machine (JVM)?

A) Compile Java source code into bytecode

B) Execute Java bytecode on different platforms

C) Manage database connections

D) Provide a graphical user interface for Java applications

Answer: B) Execute Java bytecode on different platforms

2. Which part of the Java Virtual Machine is responsible for converting bytecode to native machine code?

A) Class Loader

B) Bytecode Verifier

C) Just-In-Time (JIT) Compiler

D) Garbage Collector

Answer: C) Just-In-Time (JIT) Compiler

3. What is the purpose of the Class Loader in the Java Virtual Machine?

A) Verify the bytecode for security issues

B) Load Java classes into memory

C) Interpret the bytecode

D) Manage memory allocation for objects

Answer: B) Load Java classes into memory

4. Which memory area is used for the storage of primitive data types and reference variables in the JVM?

A) Method Area

B) Heap

C) Stack

D) Native Method Stack

Answer: C) Stack

5. What is the role of the Garbage Collector in the JVM?

A) Load and unload classes

B) Manage memory and reclaim unused objects

C) Convert bytecode to machine code

D) Verify the integrity of bytecode

Answer: B) Manage memory and reclaim unused objects

6. Which component of the JVM is responsible for checking the integrity of bytecode before it is executed?

A) Class Loader

B) Bytecode Verifier

C) Just-In-Time (JIT) Compiler

D) Memory Manager

Answer: B) Bytecode Verifier

7. What is the purpose of the Native Method Stack in the JVM?

A) Execute Java bytecode

B) Manage memory for native methods

C) Load native libraries

D) Handle exceptions

Answer: B) Manage memory for native methods

8. Which memory area in the JVM stores class metadata and constant pool information?

A) Method Area

B) Heap

C) Stack

D) Native Method Stack

Answer: A) Method Area

9. In the JVM, what does the term "Java Native Interface (JNI)" refer to?

A) A programming language for Java

B) A set of Java libraries for network communication

C) A standard programming interface for Java virtual machines

D) A framework for Java web development

Answer: C) A standard programming interface for Java virtual machines

10. Which of the following is true about the memory management in the JVM?

A) JVM relies on the operating system for memory management.

B) Memory management in the JVM is handled entirely by the programmer.

C) JVM uses automatic memory management, including garbage collection.

D) JVM does not support memory management.

Answer: C) JVM uses automatic memory management, including garbage collection.

11. What is the purpose of the PermGen (Permanent Generation) space in the JVM?

A) To store primitive data types

B) To store class metadata, constants, and interned strings

C) To manage memory for native methods

D) To execute Java bytecode

Answer: B) To store class metadata, constants, and interned strings

12. Which of the following statements is true regarding the JVM's role in platform independence?

A) JVM converts Java source code to machine code during compilation.

B) JVM ensures that Java bytecode is platform-independent by interpreting it at runtime.

C) JVM generates platform-specific bytecode.

D) JVM relies on the operating system for bytecode execution.

Answer: B) JVM ensures that Java bytecode is platform-independent by interpreting it at runtime.

13. What is the primary purpose of the JVM's Security Manager?

A) To provide encryption for Java applications

B) To manage security certificates

C) To control access to system resources by Java applications

D) To handle security vulnerabilities in the Java language

Answer: C) To control access to system resources by Java applications

14. Which thread is responsible for executing the main method in a Java application?

A) Main thread

B) Compiler thread

C) Garbage collector thread

D) Daemon thread

Answer: A) Main thread

15. What happens when the OutOfMemoryError exception occurs in the JVM?

A) The program continues execution without any interruption.

B) The JVM automatically frees up memory to avoid termination.

C) The program terminates, and an error message is displayed.

D) The JVM increases the heap size dynamically.

Answer: C) The program terminates, and an error message is displayed.

16. Which command is used to start a Java application and launch the JVM?

A) java run

B) javac

C) java start

D) java

Answer: D) java

17. What is the purpose of the -Xmx option when starting a Java application?

A) It specifies the maximum heap size for the JVM.

B) It sets the classpath for the Java application.

C) It enables the Just-In-Time (JIT) compiler.

D) It specifies the initial heap size for the JVM.

Answer: A) It specifies the maximum heap size for the JVM.

18. What is the bytecode format used by the JVM?

A) Executable code

B) Platform-specific machine code

C) High-level source code

D) Intermediate code

Answer: D) Intermediate code

19. In the context of the JVM, what is the purpose of the javac command?

A) To start the Java Virtual Machine.

B) To compile Java source code into bytecode.

C) To execute a Java application.

D) To install Java on the system.

Answer: B) To compile Java source code into bytecode.

20. Which memory area is shared among all threads in the JVM?

A) Heap

B) Stack

C) Method Area

D) Native Method Stack

Answer: C) Method Area

Multthreading mcq

1. What is the name of the method used to start a thread execution?

1. init
2. start()
3. run()
4. sleep()

2. Which two are valid constructors for Thread?

    1. Thread(Runnable r, String name)
    2. Thread()
    3. Thread(int priority)
    4. Thread(Runnable r, ThreadGroup g)
    5. Thread(Runnable r, int priority)

1. 1 and 3
2. 2 and 4
3. 1 and 2
4. 2 and 5

3. class X implements Runnable

```
{
   public static void main(String args[])
   {
      /* Missing code? */
   }
   public void run() {}
```

}

Which of the following line of code is suitable to start a thread ?

1. .Thread t = new Thread(X);
2. Thread t = new Thread(X); t.start();
3. X run = new X(); Thread t = new Thread(run); t.start();
4. Thread t = new Thread(); x.run();

4. Which two of the following methods are defined in class Thread?

1. start()
2. wait()
3. notify()
4. run()
5. terminate()

1. 1 and 4
2. 2 and 3
3. 3 and 4
4. 2 and 4

Note 2 and 3 are methods of object class

5. Which method must be defined by a class implementing the java.lang.Runnable interface?
    1. void run()
    2. public void run()
    3. public void start()
    4. void run(int priority)
6. Which will contain the body of the thread?
    1. run();
    2. start();
    3. stop();
    4. main();
7. Which method registers a thread in a thread scheduler?
    1. run();
    2. construct();

3. start();
4. register();

note Option C is correct. The start() method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread

8. Which class or interface defines the wait(), notify(),and notifyAll() methods?
    1. Object
    2. Thread
    3. Runnable
    4. Class
9. Under which conditions will a currently executing thread stop?
    1. When an interrupted exception occurs.
    2. When a thread of higher priority is ready (becomes runnable).
    3. When the thread creates a new thread.
    4. When the stop() method is called.
    1. 1 and 3
    2. 2 and 4
    3. 1 and 4
    4. 2 and 3

10. public class MyRunnable implements Runnable

```
{
   public void run()
   {
      // some code here
      }
}
```

Which of these will create and start this thread?

1. new Runnable(MyRunnable).start();
2. new Thread(MyRunnable).run();
3. new Thread(new MyRunnable()).start();
4. new MyRunnable().start();

5. What will be the output of the program?

11. What will be the output of the program?

```
class MyThread extends Thread
{
   public static void main(String [] args)
    {
       MyThread t = new MyThread();
       t.start();
       System.out.print("one. ");
      t.start();
       System.out.print("two. ");
   }
   public void run()
    {
      System.out.print("Thread ");
   }}
```

1. Compilation fails
2. An exception occurs at runtime.
3. It prints "Thread one. Thread two."
4. The output cannot be determined.

Note start cannt be called again with same instance

12. What will be the output of the program?

```
 class s1 implements Runnable
 {
     int x = 0, y = 0;
     int addX() {x++; return x;}
     int addY() {y++; return y;}
     public void run() {
     for(int i = 0; i < 10; i++)
```

```
        System.out.println(addX() + " " + addY());
}
    public static void main(String args[])
    {
        s1 run1 = new s1();
        s1 run2 = new s1();
        Thread t1 = new Thread(run1);
        Thread t2 = new Thread(run2);
        t1.start();
        t2.start();
    }
     }
```

1. Compile time Error: There is no start() method
2. Will print in this order: 1 1 2 2 3 3 4 4 5 5...
3. Will print but not exactly in an order (e.g: 1 1 11 22 2 2 3 3 3 3...)
4. Will print in this order: 1 2 3 4 5 6... 1 2 3 4 5 6...

13. What will be the output of the program?

```
class s1 implements Runnable
{
    int x, y;
    public void run()
    {
        for(int i = 0; i < 10; i++)
            synchronized(this)
            {
                x = 12;
               y = 12;            }
           System.out.println( x + " " + y);    }
    public static void main(String args[])
    {
        s1 run = new s1();
        Thread t1 = new Thread(run);
```

```
    Thread t2 = new Thread(run);
    t1.start();
    t2.start();
  } }
```

1. DeadLock
2. It print 12 12 12 12
3. Compilation Error
4. Cannot determine output.

14. Which two can be used to create a new Thread?

1. Extend java.lang.Thread and override the run() method.
2. Extend java.lang.Runnable and override the start() method.
3. Implement java.lang.Thread and implement the run() method.
4. Implement java.lang.Runnable and implement the run() method.
5. Implement java.lang.Thread and implement the start() method.

1. 1 and 2
2. 2 and 3
3. 1 and 4
4. 3 and 4

15. The following block of code creates a Thread using a Runnable target:

```
Runnable target = new MyRunnable();
Thread myThread = new Thread(target);
```

Which of the following classes can be used to create the target, so that the preceding code compiles correctly?

1. public class MyRunnable extends Runnable{public void run(){}}
2. public class MyRunnable extends Object{public void run(){}}
3. public class MyRunnable implements Runnable{public void run(){}}
4. public class MyRunnable implements Runnable{void run(){}}

16. What are the two ways to create the thread?

1. By implementing Runnable
2. By extending Thread

3. Both
4. None of these

17. What is the default thread at the time of starting the program?

1. Main thread
2. Class
3. Run
4. Thread

18. How many threads at a time can access a monitor?

1. One
2. Two
3. Ten
4. Infinite

19. When two threads are waiting on each other and can't proceed the programe is said to be in a deadlock?

    1. True
    2. False
    3. Can’t say

20. Which method waits for the thread to die ?

    1. Sleep()
    2. Wait()
    3. Join()
    4. Synchronized

21. Garbage collector thread belongs to which priority?

1. low-priority
2. high priority
3. medium priority
4. No priority

22. Daemon thread is a low priority thread which runs intermittently in the background doing the garbage collection operation for the java runtime system

1. True
2. False

23. What exception type does the following program throw? [ ]

```
public class Test {
public static void main(String[] args) {
String s = "abc";
System.out.println(s.charAt(3)); }}
```

a. Arithmetic Exception

b. ArrayIndex Out Of Bounds Exception

c. StringIndex Out Of Bounds Exception

d. Class Cast Exception

24. Given the code. What is the result when this program is executed?

```
public class SuperHotel {
   static int x[];

   static {
      x[0] = 1;
   }

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

A) ArrayIndexOutOfBoundsException is thrown

B) ExceptionInInitializerError is thrown

C) IllegalStateException is thrown

D) StackOverflowException is thrown

25. Given the code. What is the result?

```
public class Cruiser implements Runnable
{
public static void main(String[] args)
{
    Thread a = new Thread(new Cruiser());
    a.start();
    System.out.print("Begin");
    a.join();
    System.out.print("End");
}
public void run()
{
    System.out.print("Run");
}}
```

1. Compilation fails.
2. An exception is thrown at runtime.
3. "BeginRunEnd" is printed.
4. "BeginEndRun" is printed.
5. "BeginEnd" is printed.