**SelfTest**

**Chapter 1**

1. Bytecode is a highly optimized set of instructions designed to execute by what is called Java Virtual Machine.
2. Encapsulation, Inheritance, Polymorphism
3. Java programs begin their execution at the main method.
4. A variable is a named memory location. The contents of a variable can be changed during the execution of a program.
5. D
6. Single line: //, double line: /\* \*/.
7. If:
   1. If(condition)Statement;
   2. For:
   3. For(initialization;Condition,iteration)Statement;
8. Block of code starts with { and ends with }.
9. Class Moon{

Public Static void main(string args[]){

Double earthweight;

Double moonweight;

Earthweight = 165;

Moonweight = earthweight \* 0.17;

System.out.print(earthweight + “Earth-pounds is equivalent to “ + Moonweight + “ moon-pounds.”);

}

}

1. In try it yourself.
2. Syntax error
3. No, java is free from language.

**Chapter 2**

1. Java strictly specifies the range and behaviour of its primitive types to ensure portability across platforms.
2. Java’s Character type is char. Java uses Unicode rather than ACII, which are used by some other languages.
3. False, a Boolean must be either true or false.
4. System.out.println(“One\nTwo\nthree”);
5. There are two fundamental flaws in the fragment. First, sum is created each time the block defined by the for loop is entered and destroyed on exit. Thus, it will not hold its value between iterations. Attempting to use sum to hold a running sum of the iterations is pointless. Second, sum will not be known outside of the block in which it is declared. Thus, the reference to it in the println() statement is invalid.
6. When the increment operator precedes its operand, Java will perform the increment prior to obtaining the operand’s value for use by the rest of the expression. If the operator follows its operand, then Java will obtain the operand’s value before incrementing.
7. If ((b != 0) && (val /b))…
8. Byte and short are promoted to an int.
9. A cast is needed when converting between incompatible types or when a narrowing conversion is occurring.
10. Try it yourselves.
11. No
12. Yes.

**Chapter 3**

1. // Count spaces.

Public static void main(String args[])

Throws java.io.IOException{

Char ch;

Int spaces = 0;

System.out.println(“Enter a period to stop.”);

do{

ch = (char) system.in.read();

if (ch == ‘ ‘) spaces++;

} while (ch != ‘.’);

System.out.printline(“Spaces: “ + spaces);

}

}

1. If (condition)

Statement;

Else if (condition)

Statement;

Else

Statement;

1. The last else is associated with if(y>100).
2. For(int I = 1000; i> = 0; i-= 2)
3. No, I is not known outside the for loop in which it is declared.
4. A break label causes termination of its immediately enclosing loop or switch statement.

A break with a label causes control to transfer to the end of the labelled block.

1. After the break executes the after while will be displayed.
2. 0 1, 2 3, 4 5, 6 7, 8 9.
3. For (I = 1; I <100; I += i)

System.out.print(I + “ “);

1. In Netbeans.
2. An Infinite loop is a loop that run forever.
3. Yes.

**Chapter 4**

1. A class is a logical abstraction that describes the form and behaviour of an object. An object is a physical instance of the class.
2. A class is defined by using the keyword class. Inside the class statement, you specify the code and data that compromise the class.
3. Ech object of a class has its own copy of the class’ instance variable.
4. MyCounter counter; counter = new MyCounter();
5. Double myMeth(int a, int b){..}
6. A method that returns a value must return via the return stratemnt, passing back the return value in the process.
7. A constructor has the same name as its class.
8. The new operator allocates memory for an object and initialaizes it using the objects constructor.
9. Garbage collection is the mechanism that recycles unused objects so that their memory can be used.
10. The this keyword is a reference to the object on which a methos is invoked. It is automatically passed to a method.
11. Yes
12. Void

**Chapter 5**

1. Double y[] = new double [12]; double[] x = new double[12];
2. Int y[]= {1,2,3,4,5,};
3. In NetBeans.(Averages)
4. In NetBeans.(Strings)
5. The **indexof()** method finds the first occurrence of the specified substring where the **LastIndexOf()** finds the last occurrence.
6. As strange as it may look, this is a valid call to length( ):

System.out.println(“Java likes me”.length());

The output displayed is 11. **charAt()** is called in a similar fashion.

1. In NetBeans.
2. No
3. Y = x < 0 ? 10 :20;
4. It is a logical operator because the operands are type Boolean.
5. Yes.
6. >>>
7. In netbeans
8. No, the for loops in the Bubble class that perform the sort cannot be converted into for­each style loops. In the case of the outer loop, the current value of its loop counter is needed by the inner loop. In the case of the inner loop, out­of­order values must be exchanged, which implies assignments. Assignments to the underlying array cannot take place when using a for­each style loop.
9. Beginning with JDK7, the answer is Yes.
10. The name var is reserved for the use with local variable type inference.
11. Var done = false;
12. Yes, var can be the name of a variable, and No var cannot be the name of a class.
13. No, its not valid because array brackets are not allowed after var.
14. No only one variable at a time van be declared when type inference is used.
15. Var mask = 1Lk

**Chapter 6**

1. No, a private variable cannot be accessed outside of its own class.
2. Precede
3. In Netbeans.
4. Void swap(Test ob1, Test ob2) {

Int t;

T = ob1.a;

Ob1.a = ob2.a;

Ob2.a = t;}

1. No. Overloaded methods can have different return types, but they do not play a role in overload resolution. Overloaded methods must have different parameter lists.
2. In Neatbeans.
3. Shared variables are declared static.
4. A static block is used to perform any initializations related to the class, before any objects are created.
5. An inner class is a nonstatic nested class.
6. Private
7. Signature
8. Value
9. In Netbeans
10. Yes
11. Double myMeth(double … v) {//

Double myMeth(double d, double …v) {//

**Chapter 7**

1. No, a superclass does not have access to members of a sub class, yes a subclass has access to all nonprivate members of its superclass.
2. InNetbeans.
3. To prevent a subclass from having access to a superclass member, declare that member as private.
4. The super Keyword has two froms. The first is used to call a superclass constructor. The general form of this usage is : super(param-list);

The seconds form of super is used to access a superclass member.It has this general form: super.member

1. Constructors complete their execution in order of devrivation. Thus, when a Gamma object is created, the order is Alpha, Beta, Gamma.
2. When an overridden method is called through a superclass reference, it is a r=type of the object being referred to that determines which version of the method is called.
3. An abstract class contains at least one abstract method.
4. To prevent a method from being overridden, declare it as final. To prevent a class from being inherited, declare it as final.
5. Inheritance, method overriding, and abstract classes support polymorphism by enabling you to create a generalized class structure that can be implemented by a variety of classes. Thus, the abstract class defines a consistent interface that is shared by all implementing classes. This embodies the concept of “one interface, multiple methods.”
6. The Object class.
7. True
8. Final
9. Even though a B object is created, the type of myRef will be A because that is the declared return type of makeObj( ). When using local variable type inference, the inferred type of a variable is based on the declared type of its initializer. Therefore, if the initializer is of a superclass type (which is A in this case), that will be the type of the variable. It does not matter if the actual object being referred to by the initializer is an instance of a derived class.
10. In this case, the cast to B Specifies the type of the initializer, and myRef is of type B.

**Chapter 8:**