Chapter 3: User Defined Methods - Multiple Choice Questions

December 23, 2025

1 Section 1: General Methods

1. What is an advantage of using functions in programming?

   (a) Code repetition
   (b) Abstraction
   (c) Increased complexity
   (d) Slower debugging

   Ans: (b) Abstraction Functions help abstract the complexity of code by encapsulating functionality into reusable blocks. This makes code more modular and maintainable.

2. What does the term "function signature" refer to?

   (a) The body of the function
   (b) The return type of the function
   (c) The combination of the functions name and parameter list
   (d) The access specifier of the function

   Ans: (c) The combination of the functions name and parameter list A function signature uniquely identifies a function, typically including the name and parameter types, which are essential for distinguishing overloaded methods.

3. What is the correct syntax for creating an object in Java?

   (a) ClassName objectName = ClassName();
   (b) ObjectName = new ClassName();
   (c) ClassName objectName = new ClassName();
   (d) New ClassName objectName;

Ans: (c) ClassName objectName = new ClassName() This is the proper syntax for creating and initializing an object in Java using the new keyword.

4. Which of the following is an example of a non-member method?

   (a) Void display()
   (b) Static void checkSum()
   (c) Void sum()
   (d) Public void calculate()

   Ans: (b) Static void checkSum() Static methods belong to the class rather than an instance of the class, making them non-member methods in the context of objects.

5. What type of function brings about a change in its arguments and is referred to as an impure function?

   (a) Pure function
   (b) Static function
   (c) Impure function
   (d) Reference function

   Ans: (c) Impure function An impure function modifies its arguments or depends on external state, producing side effects.

6. In Java, how are objects passed to methods?

   (a) Call by Value
   (b) Call by Reference
   (c) Call by Name
   (d) Call by Assignment

   Ans: (b) Call by Reference In Java, object references are passed by value, but since the reference points to the same object, changes to the object affect the original.

7. Which of the following is a valid function prototype?

   (a) Public void calculateSum()
   (b) Public calculateSum(int a, int b)
   (c) CalculateSum(int a, int b)
   (d) Public int calculateSum(int a, int b)

Ans: (d) Public int calculateSum(int a, int b) A function prototype must include the return type, method name, and parameter list.

8. Which of these statements is true for method overloading?
   (a) Methods with the same name and identical parameter lists can be overloaded.
   (b) Methods with the same name and different parameter lists are considered overloaded
   (c) Overloaded methods must return the same data type
   (d) Overloaded methods cannot have more than two parameters

   Ans: (b) Methods with the same name and different parameter lists are considered overloaded Method overloading occurs when methods have the same name but different parameter types or counts.

9. Which method call in the following options refers to calling a member method?
   (a) ObjectName.sum()
   (b) CheckSum()
   (c) Main()
   (d) New ClassName();

   Ans: (a) ObjectName.sum() objectName.sum() because it refers to calling a member method of a class using an object. In Java, member methods are instance-specific methods, and they must be accessed through an object of the class.

10. What is the general format for calling a function that does not return a value?
    (a) Return$_typevariableName = function_name(actual_parameters)$;
    (a) Function$_name(actual_parameters)$;
    (a) Function$_name$;
    (a) Function$_name[parameters]$;

    Ans: (b) Function$_name(actual_parameters); Thisisthecorrectwaytocallafunctionthatdoesnotretur$

11. Find the odd one out
    (a) Method overloading
    (b) Polymorphism
    (c) Abstraction
    (d) Multiple forms of same

Ans: (c) Abstraction While overloading, polymorphism, and "multiple forms of the same" are related to polymorphism, abstraction deals with hiding implementation details.

12. Built-in functions are also called:

    (a) Library methods
    (b) User-defined methods
    (c) Fixed methods
    (d) All of the above

    Ans: (a) Library methods Built-in functions are predefined methods provided by programming languages or their standard libraries. These functions are part of the language's library, and they are designed to perform common tasks such as mathematical operations, string manipulations, input/output handling, etc.

13. How many return statements can a method have and how many of them should get executed for the program to run successfully?

    (a) 1, 1
    (b) Many, 1
    (c) 1, Many
    (d) Many, Many

    Ans: (b) Many, 1 A method can have multiple return statements, but only one is executed during a single call.

14. A method in which the method parameters can get modified is called a

    15. Pure method
    16. Impure method
    17. Virtual method
    18. None of the above

    Ans: (b) Impure method An impure method is a method where the parameters passed to it can be modified. This means that the method may have side effects, such as changing the state of the arguments (if they are objects or passed by reference) or modifying global variables.

    A method prototype has which of the following components?

    (a) Name of method
    (b) Return type of method
    (c) List of formal parameters
    (d) All of the above

Ans: (d) All of the above A method prototype (also called a method declaration or signature in some contexts) defines the structure of a method without providing its implementation.

A method with no result or value to return has a return type as

(a) Null
(b) Void
(c) No return type
(d) Default

Ans: (b) Void In Java (and many other programming languages), when a method does not return any value, its return type is explicitly specified as void. The void keyword indicates that the method performs some actions but does not return a result to the caller.

When a reference is passed to a method as a parameter, which of the following statement is true?

(a) The reference parameter creates a copy of the variable passed to it
(b) Reference parameter refers to the same variable passed to it
(c) The reference parameter value can get changed but passing variable value does not get changed
(d) A reference can be of simple data type

Ans: (b) Reference parameter refers to the same variable passed to it When a reference is passed to a method as a parameter, the method receives a reference to the actual object, not a copy of it. This means that any changes made to the object using the reference in the method will directly affect the original object outside the method. This behavior occurs because the reference parameter points to the same memory location as the original variable.

An operator used to invoke methods or variables using class name

(a) New
(b) Dot
(c) Relational
(d) Arithmetic

Ans: (b) Dot The dot operator (.) is used in Java to access methods or variables of a class. When used with a class name, it allows accessing static methods or static variables of the class. Similarly, when used with an object, it enables accessing instance methods or instance variables.

What would be the prototype of the below method for implementing method overloading?

int check(double x)

(a) Double check(int x, double y)

(b) Double check(double x)

(c) Both (a) and (b)

(d) Neither (a) nor (b)

Ans: (a) Double check(int x, double y) This has a different parameter list (two parameters instead of one), which makes it a valid overload. The return type (double) is irrelevant for overloading, but the parameters differ, so it satisfies the conditions for overloading.

The parameters that are used in the prototype of the method are called

(a) Actual parameters

(b) Formal parameters

(c) Virtual parameters

(d) Real parameters

Ans: (b) Formal parameters Formal parameters are the parameters listed in the method prototype or declaration. These are placeholders for the values that will be passed to the method when it is called. They are defined within the parentheses of the method signature and exist only within the scope of the method during its execution.

Which of the following java statement will compile successfully for defining a method to receive a list of 10 values?

(a) Void test-method (int a[10])

(b) Void test-method (int a[ ])

(c) Void test-method (int a)0

(d) Both (b) and (c)

Ans: (d) Both (b) and (c) In Java, when defining a method that takes an array as a parameter, the array size is not explicitly specified in the method declaration. Java does not allow specifying the size of an array in the parameter list because the size of the array is determined at runtime, not at compile time.

The number of values that a method can return is

(a) 1

(b) 2

(c) 3

(d) 4

Ans: (a) 1 In Java, a method can directly return only one value. This is because the method's return type specifies the type of the single value it can return. When a return statement is executed, the method terminates and provides that single value back to the caller.

A student writes the following code to multiply a variable z by 5. He has written the following statement, which is incorrect:
z * 5;
What will be the correct statement?
A. z *= 5;
B. z = z * 5;
C. z * 5;

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans: (c) Both A and B The given statement z * 5; is incorrect because it performs a multiplication operation but does not store the result anywhere. In Java, such a statement has no effect since the result of the multiplication is not assigned back to the variable or used elsewhere.

A student writes the following code to divide a variable a by 4. He has written the following statement, which is incorrect:
a = /4;
What will be the correct statement?
A. a /= 4;
B. a = a / 4;
C. a = /4;

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans: (c) Both A and B The given statement a = /4; is incorrect because it is not valid syntax in Java. The / operator must be used between two operands, and the statement must assign the result to a variable correctly.

A student writes the following code to increment a variable n by 1. He has written the following statement, which is incorrect:
n + 1;
A. n++;
B. n += 1;
C. n = n + 1;

(a) Only A

(b) Only C

(c) All A, B, and C

(d) None of the above

Ans: (c) All A, B, and C The statement written by the student, n + 1;, is incorrect because it simply evaluates the expression n + 1 but does not assign the result back to the variable n. All three forms increment n by 1.

A student writes the following code to subtract 7 from a variable p.
He has written the following statement, which is incorrect:
p = 7-;
What will be the correct statement?
A. p -= 7;
B. p = p - 7;
C. p = -7;

(a) Both A and B

(b) Only B

(c) Only C

(d) None of the above

Ans: (a) Both A and B The given statement p = 7-; is incorrect because it is not valid syntax.

2 Section 2: Constructors

1. A special method that initializes values to data members is called

   2. Constructor
   3. Method
   4. Function
   5. All of the above

Ans: (a) Constructor A constructor is a special method in object-oriented programming used to initialize the values of data members when an object is created. It is automatically called when the object is instantiated.

Is the following statement true/false?
Access specifiers can be used for defining constructors.

(a) True

(b) False

(c) May be

(d) May not be

Ans: (a) True Access specifiers (public, private, and protected) can indeed be used for defining constructors in object-oriented programming. The access specifier determines the level of accessibility of the constructor.

Which of the following is a return type of a constructor?

(a) Void
(b) Int
(c) Null
(d) None of the above

Ans: (d) None of the above A constructor does not have a return type, not even void. Constructors are special methods in object-oriented programming that are called automatically when an object is created. They initialize the object's properties but do not return any value.

If a constructor is declared as private, how many instances can be created for the class?

(a) 1
(b) 2
(c) Many
(d) 0

Ans: (a) 1 A private constructor restricts object creation, typically allowing only one instance via a factory method or singleton pattern.

How would the compiler/interpreter behave if there is no constructor defined in the class?

(a) No error. Default values will be initialized to members
(b) Error
(c) Impossible
(d) Program hangs

Ans: (a) No error. Default values will be initialized to members Java always provides a default constructor if no constructor is defined. Primitive types are initialized to their default values (0 for numeric types, false for boolean, null for references).

State the output of the below code:

class Abc int a; String c; public static void main() Abc ob = new Abc(); System.out.print(ob.a + " " + ob.c);

(a) 1 * 1
(b) 0 0
(c) 0
(d) 0 null

Ans: (d) 0 null Instance variables of a class in Java are automatically initialized to their default values if not explicitly assigned. Default value for int is 0. Default value for String (a reference type) is null.

State the output of the below code:

class Abc int a; String c; Abc(int x, String y) a = x; c = y; public static void main() Abc ob = new Abc(2, 1"); System.out.print(ob.a + " + ob.c);

(a) ?null

(b) error

(c) nullnull

(d) 2 1

Ans: (d) 2 1 The parameterized constructor initializes a and c with 2 and "1", respectively.

A constructor can only be written $_{i}^{naclass.}$

At the beginning

At the end

Anywhere

None of the above

Ans: (c) anywhere A constructor can be written anywhere within a class. Its placement in the code does not matter because it is identified by its name matching the class name and having no return type. The compiler recognizes it based on its signature.

An object is allocated the memory when the execution of $_{i}^{scompleted.}$

Main method

Constructor

Program

All of the above

Ans: (b) Constructor An object is allocated memory when its constructor finishes execution. The constructor initializes the object, setting up its properties and resources, and once completed, the object is ready for use in memory.

A constructor with no parameters is called .

Non-parameterized constructor

Parameterized constructor

None

Error

Ans: (a) Non-parameterized constructor A constructor with no parameters is called a non-parameterized or default constructor. It initializes the object with default values and does not require arguments during object creation.

A student tries to create a constructor in a class Sample. The constructor is written as: void Sample() System.out.println("Constructor called"); What is incorrect in the above statement, and what will be the correct statement?

(a) Only A
(b) Only B
(c) All the three
(d) Both A and B

Ans: (a) Only A The provided constructor is incorrectly defined as a method (with a return type). Constructors should not specify a return type.

What is the purpose of a constructor in a class?

(a) To define class methods
(b) To initialize class data members
(c) To destroy objects
(d) To handle exceptions

Ans: (b) To initialize class data members A constructor is specifically designed to initialize the data members of a class when an object is created.

Which statement about a constructor is TRUE?

(a) It must have a return type.
(b) Its name must match the class name.
(c) It must be explicitly called by the user.
(d) It can have a different name than the class.

Ans: (b) Its name must match the class name. A constructor's name must exactly match the class name to ensure automatic invocation during object creation.

What is a non-parameterized constructor?

(a) A constructor that does not return anything.
(b) A constructor that does not accept parameters.
(c) A constructor with only default values.
(d) A constructor with a private access specifier.

Ans: (b) A constructor that does not accept parameters. Non-parameterized constructors do not take any arguments and usually set default values for data members.

What happens if no constructor is defined in a class?

(a) The program throws an error.
(b) A default constructor is automatically provided.
(c) The class cannot be instantiated.
(d) The object is created without initialization.

Ans: (b) A default constructor is automatically provided. Java provides a default constructor that initializes member variables to default values if no constructor is explicitly defined.

Which access specifier is most commonly used for constructors?

(a) Private
(b) Public
(c) Protected
(d) Default

Ans: (b) Public Constructors are typically declared public to allow objects of the class to be created from any other class. If private, the class can only be instantiated from within the class itself, restricting object creation to controlled scenarios. Public access ensures the class can be widely used.

What is the primary function of a parameterized constructor?

(a) To return a value

(b) To set default values

(c) To accept arguments for initialization

(d) To invoke another constructor

Ans: (c) To accept arguments for initialization Parameterized constructors are used to initialize class data members with specific values passed as arguments during object creation, offering flexibility and customization.

What is constructor overloading?

(a) Using multiple constructors with the same name but different access specifiers.

(b) Defining multiple constructors with different parameter lists.

(c) Calling one constructor from another.

(d) Overriding a constructor in a subclass.

Ans: (b) Defining multiple constructors with different parameter lists. Constructor overloading allows a class to have multiple constructors with varying numbers or types of parameters, providing different initialization options for objects.

Which of the following is TRUE about formal parameters?

(a) They are values passed to a constructor during object creation.

(b) They are variables declared in the constructor definition.

(c) They are always initialized to default values.

(d) They are only used in default constructors.

Ans: (b) They are variables declared in the constructor definition. Formal parameters are the placeholders defined in the constructor's parameter list, receiving values passed by actual parameters during object creation.

What is the default value of a member variable in Java?

(a) null for all types

(b) 0 for numeric types, null for objects

(c) User-defined default values

(d) Compiler-defined random values

Ans: (b) 0 for numeric types, null for objects In Java, member variables are automatically initialized with default values: 0 for numeric types, false for boolean, and null for object references, unless explicitly initialized.

Which of these correctly initializes an object using a default constructor?

(a) MyClass obj = new MyClass(5);
(b) MyClass obj = new MyClass();
(c) MyClass obj;
(d) obj = MyClass();

Ans: (b) MyClass obj = new MyClass(); A default constructor takes no arguments. The syntax MyClass obj = new MyClass(); creates an object using the default constructor.

Which of these is an actual parameter?

(a) Variables declared in the constructor definition.
(b) Variables passed to a constructor during object creation.
(c) Parameters used to call a method.
(d) Variables that define the constructor.

Ans: (b) Variables passed to a constructor during object creation. Actual parameters are the values or variables passed to a constructor during object creation, which are then copied to formal parameters.

What is a formal parameter in a constructor?

(a) A variable declared outside the constructor.
(b) A placeholder variable declared in the constructor definition.
(c) A variable used to call a method.
(d) A parameter used to call another constructor.

Ans: (b) A placeholder variable declared in the constructor definition. Formal parameters are variables defined in the constructor's parameter list and used to receive the values passed by actual parameters.

How do you define a parameterized constructor?

(a) By providing default values to all variables.
(b) By defining a constructor with parameters to accept values.
(c) By using a special access specifier.
(d) By defining a constructor without any parameters.

Ans: (b) By defining a constructor with parameters to accept values. A parameterized constructor accepts arguments that are used to initialize the data members of a class with specific values during object creation.

What happens when you call a constructor explicitly in Java?

(a) It initializes the object multiple times.
(b) It throws an error.
(c) The constructor cannot be explicitly called.
(d) It overrides the default constructor.

Ans: (c) The constructor cannot be explicitly called. In Java, constructors are automatically invoked during object creation and cannot be explicitly called like methods.

Which of these is an example of constructor overloading?

(a) A class having multiple constructors with the same name and parameter lists.
(b) A class having multiple constructors with different parameter lists.
(c) A class having a single constructor.
(d) A class having private constructors only.

Ans: (b) A class having multiple constructors with different parameter lists. Constructor overloading refers to having multiple constructors in a class, differing in the number or types of parameters.

Which constructor is automatically invoked if none is defined in the class?

(a) Parameterized constructor
(b) Default constructor
(c) Private constructor
(d) Static constructor

Ans: (b) Default constructor If no constructor is defined in a class, the compiler provides a default constructor that initializes the object with default values.