QUESTION BANK 1

QUESTION BANK

1

Declarations, Initialization, and Scoping

CERTIFICATION OBJECTIVES

1.1 Develop code that declares classes (including abstract and all forms of nested classes), interfaces, and enums, and includes the appropriate use of package and import statements (including static imports).

1.2 Develop code that declares an interface. Develop code that implements or extends one or more interfaces. Develop code that declares an abstract class. Develop code that extends an abstract class.

1.3 Develop code that declares, initializes, and uses primitives, arrays, enums, and objects as static, instance, and local variables. Also, use legal identifiers for variable names.

1.4 Develop code that declares both static and non-static methods, and - if appropriate - use method names that adhere to the JavaBeans naming standards. Also develop code that declares and uses a variable-length argument list.

1.5 Given a code example, determine if a method is correctly overriding or overloading another method, and identify legal return values (including covariant returns), for the method.

1.6 Given a set of classes and superclasses, develop constructors for one or more of the classes. Given a class declaration, determine if a default constructor will be created, and if so, determine the behavior of that constructor. Given a nested or non-nested class listing, write code to instantiate the class.

QUESTION 1.1

Q 1: **Imagine you define the MyInterface interface as shown in the following code snippet:**

```
package com.kogent;
interface MyInterface {
        //complete the code here
        final int mynum = 70;
}
```

Which of the following options will lead to compilation error?

A. public final void myMethod(); B. protected void myMethod();
C. public void myMethod(); D. private abstract void myMethod();

A 1: **The correct options are A, B, and D.**

Explanation: The following are the invalid method declarations in an interface:

- public final void myMethod();
- private abstract void myMethod();
- protected void myMethod();

A method within an interface cannot be marked as final, as final modifier disallows overriding of a method and the methods declared within an interface do not have method body. Therefore, the final modifier cannot be used as method declarations within an interface.

Within an interface, you cannot declare a method using both private and abstract as the private class members can be accessed only within the class in which they are declared. Therefore, a method declared within an interface is declared as public and not as private or protected. As a result, the option A, B, and D will lead to compilation error and are the correct answers.

QUESTION 1.2

Q 2: **Imagine you are a software developer and write the following program:**

```
package com.kogent;
public class VarargsExample {
        public static void displayNames(String... names) {
                for (String mynames:names) {
                        System.out.print(mynames + " ");
                }
        }
        public static void main(String args[]) {
                displayNames("Alex","Richard","John");
        }
}
```

What will be the output after compiling and executing the preceding program?

A. The program leads to compilation error.
B. The program compiles successfully and displays "Alex Richard John" as output.
C. The program compiles successfully and leads to runtime exception.
D. The program compiles successfully but does not display anything as output.

A 2: **The correct answer is B.**

Explanation: The preceding program displays "Alex Richard John" as output as it demonstrates the variable argument list concept. Therefore, option B is correct.

QUESTION 1.3

Q 3: **Imagine while preparing for your SCJP examination, you created the following program:**

```
package com.kogent;
class Ques7 {
        int eval(int[]...vars) {
                int sum=0, b, c;
                for(b = 0; b<vars.length; b++) {
                        for(c=0;c<vars[b].length; c++) {
                                sum += vars[b][c];
                        }
                }
                return(sum);
        }
```

```
        public static void main(String args[]) {
                Ques7 varargs = new Ques7();
                int sum =0;
                sum = varargs.eval(new int[]{10,20,30,40}, new int[]{40,50,60});
                System.out.println("The sum of the numbers is:" + sum);
        }
}
```

What will happen during compilation and execution of your program?

A. The program will compile and display "The sum of the numbers is: 250"as output.
B. The program will compile and display 25 as output.
C. The program will not compile due to invalid declaration of integer variable arguments.
D. The program will generate the runtime exception.

A 3: **The correct option is A.**

Explanation: The code will compile and execute successfully displaying "The sum of the numbers is: 250"as output because the eval(int[]…vars) method is declared with a variable argument list as its parameter. In the code, a two dimensional array is declared to evaluate the sum and the eval method is invoked from the main method.

QUESTION 1.4

Q 4: **Imagine you write the following program:**

```
package com.kogent;
enum Students{Suchita, Deepak, Vikash, Charu, Mahtab}
class Ques10 {
        public static void main(String args[]) {
        Students student;
        student = Students.Vikash;
        switch(student) {
        case Suchita: System.out.println("My name is Suchita");
        break;
        case Deepak: System.out.println("My name is Deepak");
        break;
        case Vikash: System.out.println("My name is Vikash");
        break;
        case Charu: System.out.println("My name is Charu");
        break;
        case Mahtab: System.out.println("My name is Mahtab");
        break;
        }
        }
}
```

What will be the output after the preceding program is compiled and executed?

A. The program will compile successfully and execute by displaying the output, My name is Vikash.
B. The program will lead to compilation error as the enum Students is declared outside the class.
C. The program will compile successfully and execute by displaying the output, My name is Deepak.
D. The program will lead to runtime error.

A 4: **The correct option is A.**

Explanation: The enumeration can be declared within or outside a class and so the above program will compile successfully and display My name is Vikash as the output. Therefore, option A is the correct option and B, C, and D are invalid options.

QUESTION 1.5

Q 5: **Imagine you write the following program:**

```
package com.kogent;
class Ques11 {
        public static void main(String args[]) {
                byte b = 12;
                int y = b;
                b = b + 10;
                System.out.println(b);
        }
}
```

What will be the output after the preceding program is compiled and executed?

A. The program will compile, execute, and display 22 as output.
B. The program will lead to compile time error as explicit casting is required in the line, b = b + 10.
C. The program will compile, execute, and display 12 as output.
D. The program will lead to compile time error as explicit casting is required in the line, int y =b.

A 5: The correct option is B.

Explanation: When you compile the above program, a compile time error occurs in the line, b = b + 10 as explicit casting is required to assign an int value to a byte type. In the preceding program the byte type variable is declared and initialized to the value 12. Then the value of byte variable is assigned to the int variable, y as assigning byte type value to an int variable is possible. However the compilation error of loss of precision will occur while incrementing the byte type value with 10, i.e. int value. Therefore A, C, and D are incorrect options.

QUESTION 1.6

Q 6: Imagine you write the following lines of code:

```
package com.kogent;
class Ques12 {
        public static void main(String args[]) {
                int x = 201;
                myMethod(x++);
                System.out.println(x);
        }
        static void myMethod(int x) {
                x %= 10;
                System.out.println(x);
        }
}
```

What will be output of the above program after compilation and execution?

A. The program will compile successfully and execute displaying 1 and 202 as output.
B. The program will compile successfully and execute displaying 2 and 202 as output.
C. The program will compile successfully and execute displaying 1 and 201 as output.
D. The program will compile successfully and execute displaying 1 and 1 as output.

A 6: The correct option is A.

Explanation: After compiling and executing the above program 1 and 202 will be displayed as output. In the given program, the Ques12 class contains two static methods, the main() and myMethod(). Each of the static method defines a local variable x, having same name. When the program executes the myMethod() method is invoked and the value, 201 is passed as an argument to the myMethod() method which is assigned to its local variable, x. Then the compound operator performs the modulus operation and the resultant value (1) is displayed. Finally the value of the local variable within the main() method, i.e 202 (after increment) is displayed.

QUESTION 1.7

Q 7: Consider the following various array declarations:

```
int [] ar1, arr2[];
int[][] arr3;
int[] arr4[], arr5[];
Which of the following options are true?
arr2 = arr3;
arr2 = arr4;
arr1 = arr2;
arr4 = arr1;
```

A 7: The correct answer is A and B.

Explanation: There is a difference between int[] i; and int i[]; although in both the cases i is an array of integer values. Therefore the correct option is A and B as they are the declarations of the integer. The basic difference is that if you declare multiple variables in the same statement such as int[] i, j; and int i[], j; then it implies that i and j are not of the same type.

QUESTION 1.8

Q 8: Imagine you write the following lines of code in your program:

```
package com.kogent;
class QuesSuper
{
        public int mynum=0;
        public QuesSuper(String str) {
                mynum=10;
        }
}
public class QuesSub extends QuesSuper {
        public QuesSub(String str) {
                mynum=20;
        }
        public static void main(String args[]) {
                QuesSub sub= new QuesSub("Suchita");
                System.out.println(sub.mynum);
        }
}
```

What will be the output after the preceding program is compiled and executed?

A. The program will compile successfully and 20 will be displayed as output.
B. The program will lead to compile time error.
C. The program will compile successfully and 10 will be displayed as output.
D. The program will compile successfully and 0 will be displayed as output.

A 8: The correct option is B.

Explanation: In the preceding program, the main() method of the QuesSub class invokes the constructor while creating an instance of the QuesSub class. Now before invoking constructor of the subclass constructor of the superclass is invoked. If constructor of the superclass is explicitly declared, then it is invoked else the default constructor is invoked. However in the above program constructor is explicitly declared with a parameter in QuesSuper and the constructor of QuesSub class is not passing any value to it. Therefore the above program will lead compilation error and hence the correct option is B.

QUESTION 1.9

Q 9: Which among the following are valid declarations?

```
int num1, num2, num3; num1 = num2= num3= 10;
int num1, num2, num3 =10;
int num1= 10 = num2 =num3
int num1 = 10= num2= num3;
```

A 9: The correct options are A and B.

Explanation: The multiple variables can be declared and assigned the values in either of the following ways:

```
int num1, num2, num3; //declaring the integer numbers
num1=num2=num3 = 10; //assigning values to the variables
```

or

```
int num1,num2,num3 = 10; //declaring and assigning values simultaneously
Therefore the correct options are A and B.
```

QUESTION 1.10

Q 10: Imagine you have declared the Ques20 class by using the following program:

```
package com.kogent;
public class Ques20 {
        public String name;
}
```

Now you realized that to make the name variable as read only for the other classes. Which of the following options are correct to mark the name variable as read only?

A. You can mark the name variable as private.

B. You can mark the name variable as private and provide the public method getName() which will return its value.
C. You can mark the name variable as protected.
D. You can mark the name variable as static and provide the public static method getName() which will return its value.

A 10: The correct option is B.

Explanation: In Java, the standard way to provide the read only access to the variables is to mark the variable as private and provide a public method returning its value.

QUESTION 1.11

Q 11: Which of the following options are true if you want to access the fields or methods of class B through the instance of class A, provided that the member has no access specifier?

A. The class B must be a subclass of the class A.
B. The class A and B both must be within the same package.
C. The class B must be a superclass of the class A.
D. The class A and B may not be in the same package but class B must also be a subclass of the class A.

A 11: The correct option is B.

Explanation: When you do not specify the access specifier by default it is the default access implying that the classes within the same package can be accessed by each other. Therefore, option B is the correct answer as if the class B is a subclass and not in the same package then it will not be accessed. Therefore, the option A is incorrect.

QUESTION 1.12

Q 12: Which of the following statements are true based on the use of modifiers?

A. Local variables can be declared either static or transient.
B. The visibility of the local variables cannot be specified.
C. By default the variable is accessible within the same class and subclass of the super class.
D. The visibility of the local variables is default.

A 12: The correct option is B.

Explanation: The local variables cannot be marked as transient, volatile, and static, Correct option is B and the local variable does not have any accessibility as they are accessible only from the block in which they are declared.

QUESTION 1.13

Q 13: Which of the following are valid declarations of the main () method?

A. static main(String args[]){ }
B. public static String main(String args[]) {… }
C. public static void main(String args[]) {….}
D. final static void main(String args[]) {….}

A 13: The correct option is C.

Explanation: The following is a valid declaration of the static method:

```
public static void main(String args[]) {
//implementation of the main method
}
```

Therefore, the correct option is C since the return type of the main method is void and is declared as static

QUESTION 1.14

Q 14: Imagine you need to handle the records of multiple students and declaring a separate variable and then assigning the values will become a tedious task. Therefore, you write the following program to implement the concept of arrays which has simplified your task:

```
package com.kogent;
        public class Ques30 {
                public static void main(String args[]) {
                String[][][] arr  = {
                { { "Suchita", "Vikash" , "Deepak"}, { "Charu", null, "Shikha"} },
                { {"Shalini"}, {null} },
```

```
                    {{"Hemal"}},
                    { { "Santosh", "Manish"}, {} }
                    };
                    System.out.println(arr[0][1][1]);
            }
    }
```

What will be the output after compilation and execution of the preceding program?

A. The program will throw the runtime exception.
B. The program will throw ArrayIndexOutOfBoundsException.
C. The program will display null.
D. The program will compile successfully but it will not display anything.

A 14: The correct option is C.

Explanation: The program deals with three dimensional array and will display null as following is the structure of elements assigned to the arr array.

arr[0][0][0] = Suchita
arr[0][0][1] = Vikash
arr[0][0][2] = Deepak
arr[0][1][0] = Charu
arr[0][1][1] = null
arr[0][1][2] = Shikha
arr[1][0][0] = Shalini
arr[1][1][0] = null
arr[2][0][0] = Hemal
arr[3][0][0] = Santosh
arr[3][0][1] = Manish

Therefore, if you try to print the value of the arr[0][1][1] element of the arr array, then the null value will be displayed.

QUESTION 1.15

Q 15: Which of the following is the correct higher to lower order of restrictiveness for access specifiers?

A. public> default(within the package)> protected> private
B. private> default(within the package)> protected> public
C. private> protected> default(within the package)> public
D. protected> default(within the package)> private> public

A 15: The correct option is B.

Explanation: The private class members can be accessed only within the class in which they are declared and therefore the private access specifier is highly restrictive. Moreover, the members with default accessibility are accessible within the class in which they are declared and by the classes belonging to the same package. In addition, the protected members are also accessible from subclasses and therefore the protected access specifier is less restrictive as compared to the default accessibility.

QUESTION 1.16

Q 16: In Java few keywords are reserved which you cannot use while declaring a class member. Which of the following is not a keyword in Java?

A. switch B. extends C. assert D. String

A 16: The correct option is D.

Explanation: Among the preceding options, String is the class name and not the Java keyword. Therefore, the correct option is D.

QUESTION 1.17

Q 17: Imagine you need to declare an abstract method, which of the following is the valid declaration of an abstract method?

A. class Vehicle { abstract void move(); }

B. class Vehicle { abstract void move(); {….} }
C. abstract class Vehicle { abstract void move(); }
D. abstract class Vehicle { abstract void move(); {….} }

A 17: The correct option is C.

Explanation: An abstract method is a method which is being declared by using the abstract modifier and does not have the method body or method implementation. The implementation of an abstract method specified in the subclass provided that the subclass is not an abstract class. Therefore the option B and D are invalid as these options provide the implementation of an abstract method. Moreover an abstract method must be declared in an abstract class and so options A and B are also invalid as the class defining the move() abstract method has not been declared as abstract. As a result, the correct option is C.

QUESTION 1.18

Q 18: Imagine you want to clear your concept of nested classes and so you create a program containing nested and static classes. Consider that you have created the following program:

```
package com.kogent;
public class Ques43 {
        public static void main(String args[]) {
                TestOuter o = new TestOuter();
                TestOuter.TestInner i = o.new TestInner();
                TestOuter.TestStaticInner inner = new TestOuter.TestStaticInner();
        }
}
class TestOuter {
        static int num1 = 100;
        TestOuter() {
                System.out.print("Welcome to the outer class" + " ");
        }
        class TestInner {
                TestInner() {
                        System.out.print(TestOuter.num1 + " ");
                }
        }
        static class TestStaticInner {
                static int staticnum = 200;
                TestStaticInner()       {
                        System.out.print(staticnum + " ");
                }
        }
}
```

What will be the output after you compile and execute the preceding program?

A. The program compiles successfully and displays "Welcome to the outer class 100 200" as output.
B. The program compiles successfully and displays "Welcome to the outer class 200 100" as output.
C. The program compiles successfully and displays "Welcome to the outer class 100" as output.
D. The program compiles successfully and displays "Welcome to the outer class 200" as output.

A 18: The correct option is A.

Explanation: The first statement in the main method creates an instance of the TestOuter class and then the second statement instantiates the TestInner class. The instantiation of the TestInner class is done by using the outer class instance as the TestInner class is a non-static class. Finally an instance of the TestStaticInner class is created without using the instance of the TestOuter class as the TestStaticInner class is a static class. As a result the correct option is A.

QUESTION 1.19

Q 19: Imagine that you are a Java programmer in the ABC Company and create the following program:

```
package com.kogent;
public class Ques45 {
        public void myMethod1() {
                static int num1=100;
                final int num2=200;
                System.out.println("The value of first variable is " + num1);
                System.out.println("The value of second variable is " + num2);
        }
        public void myMethod2() {
                int arr[] =  new int[2];
```

```
                System.out.println(arr[arr.length-1]);
        }
        public static void main(String args[]) {
                new Ques45().myMethod1();
                new Ques45().myMethod2();
        }
}
```

What will be the output after you compile and execute the preceding program?

A. The program will lead to compilation errors as static variables cannot be declared within methods.

B. The program will compile successfully and display "The value of first variable is 100" and "The value of second variable is 200", as output.

C. The program will compile successfully and lead to the ArrayIndexOutOfBoundsException exception during runtime.

D. The program will lead to compilation errors as the object arr is not initialized.

A 19: The correct option is A.

Explanation: The static variables cannot be declared within a method and therefore the correct option is A. However the final variables can be declared within a method and all array elements in an integer array are by default initialized to 0. As a result the correct option is A.

QUESTION 1.20

Q 20: Imagine during your career as a Java developer your mentee creates the following program:

```
package com.kogent;
public class Ques46 {
        private static int num1 = 100;
        private int num2 = 200;
        public static void myMethod1() {
                num1 = 300;
                num2 = 400;
                System.out.println(num1 + ",” + num2);
        }
        public static void myMethod2() {
                num1 = 300;
                Ques46.num2 = 400;
        }
        public void myMethod3(){
                num1 = 300;
                num2 = 400;
        }
        public void myMethod4() {
                Ques46.num1 = 300;
                num2 = 400;
        }
        public static void main(String args[]) {
                Ques46 q = new Ques46();
                q.myMethod1();
        }
}
```

Now you need to analyze the preceding program and give a feedback to your mentor with explanation. Therefore, which of the following statements you can provide as a feedback to your mentee?

A. The program will compile successfully.

B. The program will lead to compilation error as the non-static variables cannot be referenced from a static context.

C. The program will compile successfully and lead to runtime error.

D. The program will compile successfully and display "300,400" as output.

A 20: The correct option is B.

Explanation: The correct answer is B as it is not possible to access the non-static variable within the static method. Therefore options A, C, and D are incorrect.

QUESTION 1.21

Q 21: Imagine you are working a Java programmer in the ABC Company and write the following program:

```
package com.kogent;
public class Ques48 {
```

```
        public static void main(String[] args) {
                Vehicle v = new Car();
                System.out.print(v.getVehicle().getClass().getName() + " ");
                System.out.print(v.getVehicle().getName());
        }
}
class Vehicle {
        public Vehicle getVehicle() {
                return this;
        }
        public String getName() {
                return "Vehicle";
        }
}
class Car extends Vehicle {
        public Vehicle getVehicle() {
                return this;
        }
        public String getName() {
                return "Car";
        }
}
```

What will be the output after you compile and execute the preceding program?

A. The program will lead to compilation errors as the Car class overloads the getVehicle method by changing its return type.

B. The program will compile successfully and display "com.kogent.Car Car" as output.

C. The program will compile successfully but lead to runtime error.

D. The program will lead to compilation error at Vehicle v = new Car();

A 21: The correct option is B.

Explanation: The program will not lead to any compilation error as the getVehicle() and getName() are examples of method overriding and not overloading. Therefore the options A and D are incorrect. The code v.getVehicle() will return an instance of the Car class and its class name is com.kogent.Car, therefore the correct option is B.

QUESTION 1.22

Q 22: Imagine while practicing the concept of primitive variables in Java, you came across the following program:

```
package com.kogent;
public class Ques51 {
        public static void main(String args[]) {
                Ques51 q = new Ques51();
                q.method(30);
                byte b = 3;
                q.method(b);
        }
        public void method(Integer i) {
                System.out.print("Integer value is: " + i + " ");
        }
        public void method(short s) {
                System.out.print("Short value is: " + s + " ");
        }
        public void method(byte t) {
                System.out.print("Byte value is: " + t + " ");
        }
        public void method(int num) {
                System.out.print("Int value is: " + num + " ");
        }
}
```

What will be output of the preceding program?

A. The program will display "Int value is: 30 Byte value is: 3" as output.

B. The program will display "Integer value is: 30 Byte value is: 3" as output.

C. The program will display "Int value is: 30 Short value is: 3" as output.

D. The program will display "Integer value is: 30 Short value is: 3" as output.

A 22: The correct option is A.

Explanation: In the preceding program, widening is preferred over boxing and therefore while invoking the method() method through the argument 30, it will be widened to call the method(int num).

QUESTION 1.23

Q 23: Imagine you are working in the ABC Company and you are assigned a project with a team. Being a team leader you need to analyze the programs created by your team members. While analyzing the programs, you came across the following program:

```
package com.kogent;
public class Ques52 {
        public static void main(String args[]) {
                Ques52 q = new Ques52();
                q.myMethod(10,20);
                q.myMethod(new long[]{});
                q.myMethod(new int[]{10,20});
        }
        void myMethod(short s1, short s2) {
                System.out.println("short");
        }
        void myMethod(int i1, int i2) {
                System.out.println("int");
        }
        void myMethod(int  ...args) {
                System.out.println("intargs");
        }
}
```

Which of the following statements are justified in the context of the preceding program?

A. The program will compile successfully and display "int intargs intargs" as output.
B. The program will lead to compilation error.
C. The program will compile successfully but lead to runtime exception.
D. The program will display "short intargs intargs" as output.

A 23: The correct option is B.

Explanation: The program leads to compilation errors as while invoking the q.myMethod(new long[]{}) method, implementation of the myMethod() method does not have the long array type argument. Therefore the option B is the correct answer.

QUESTION 1.24

Q 24: Imagine you write the following program while understanding the concept of primitive variables:

```
package com.kogent;
public class Ques53 {
        public static void main(String args[]) {
                System.out.println(myMethod(myMethod(new int[]
                {10,20}),myMethod(10,20)));
        }
        static int myMethod(int num1, int num2) {
                return 10;
        }
        static int myMethod(int... args) {
                return 20;
        }
}
```

What will be output of the preceding program?

A. The program will compile successfully and display 10 as output.
B. The program will lead to compile time error as the myMethod with int[], int[] argument is not defined.
C. The program will compile successfully but lead to runtime exception.
D. The program will compile successfully and display 20 as output.

A 24: The correct option is A.

Explanation: In the preceding program, the code myMethod(new int[] {10,20}) invokes the myMethod(int …args) which returns 20. Moreover the code myMethod(10,20) invokes the myMethod(int num1, int num2)

method which returns 10. Finally the myMethod(myMethod(new int[] {10,20}), myMethod(10,20)) method becomes myMethod(20,10) which invokes myMethod(int num1, int num2), thereby returning 10, which is displayed. Therefore the correct answer is A.

QUESTION 1.25

Q 25: Imagine you are a Java programmer and you have created the following program:

```
package com.kogent;
public class Ques59 {
        public static void main(String[] args) {
                System.out.println(myMethod(new double[]{10, 20, 30}));
                System.out.println(myMethod(new Double[]{10d, 20d, 30d}));
                System.out.println(myMethod(10, 20, 30));
                System.out.println(myMethod());
        }
        static double myMethod(double ... args) {
                double total = 0;
                for (double temp : args)  {
                        total += temp;
                }
                return total;
        }
        static double myMethod(Double ... args) {
                double total = 2;
                for (double temp : args) {
                        total *= temp;
                } return total;
        }
}
```

What will be output of the preceding program?

A. The program will lead to compilation error.
B. The program will compile successfully and display "60.0 12000.0 60.0" as output.
C. The program will compile successfully but lead to runtime error.
D. The program will compile successfully and display "60.0 60.0 12000.0" as output.

A 25: The correct option is A.

Explanation: The program will lead to compilation error while invoking the myMethod() method without any argument. This is because of the ambiguity caused due to declarations of the myMethod taking primitive double variable argument and no declaration of the MyMethod without any argument..

QUESTION 1.26

Q 26: Imagine you have created the following program to have a better understanding for enums.

```
package com.kogent;
enum MyEnum {Suchita, Vikash, Deepak };
public class Ques61 {
        public static void main(String args[]) {
        for (MyEnum en: MyEnum.values())
        System.out.print(en + " ");
        }
}
```

What will be the output of the preceding program?

A. The program will compile successfully and print "Suchita Vikash Deepak" as output.
B. The program will lead to compilation error in the line for(MyEnum en: MyEnum.values())
C. The program will compile successful but lead to runtime error.
D. The program will not compile as the enum cannot be declared outside the class.

A 26: The correct option is A.

Explanation: The preceding program declares the enum outside the class which is legal and therefore the option D is incorrect. The program will compile successfully and print the values assigned in the MyEnum. Therefore the correct option is A.

QUESTION 1.27

Q 27: Imagine being a Java programmer you write the following program:

```
package com.kogent;
public class Ques63 {
        String str;
        int i=10;
        static void myMethod() {
                System.out.println("The value of String  variable is" + new
                Ques63().str.length());
        }
        public static void main(String args[]) {
                myMethod();
        }
}
```

Which of the following statements are true in the context of the preceding program?

A. The program will lead to compilation error as a non–static variable cannot be accessed from static context.

B. The program will compile successfully but lead to runtime exception.

C. The program will lead to compile time error as the String variable str is not assigned a value.

D. The program will compile successfully and print 4 as output.

A 27: The correct option is B.

Explanation: The String variable str is not assigned a value so the default value null will be assigned. The myMethod()is a static method and the str variable is a non static variable, therefore the Ques63 class reference is used to access the str variable. The compiler will compile the program successfully as the default value null will be assigned to the str variable. However during execution the NullPointerException exception occurs as the length function is invoked on the str variable which is not initialized.

QUESTION 1.28

Q 28: Imagine you are a faculty in an institute and you have explained the concept of Inner classes to the students. While practicing the students created the following program and you were asked to analyze the program:

```
package com.kogent;
public class Ques64 {
        void myMethod(){
                System.out.println("Welcome to the world of programming");
        }
        class MyNest {
                public static void main(String args[]) {
                        Ques64 q = new Ques64();
                        q.myMethod();
                }
        }
}
```

What will be output of the preceding program?

A. The program will compile successfully and print "Welcome to the world of programming" as output.

B. The program will compile successfully but lead to runtime error.

C. The program will lead to compilation error.

D. The program will compile successfully but no output is displayed.

A 28: The correct option is C.

Explanation: The inner classes cannot have static declarations and so the preceding program will lead to compilation error. Therefore the correct option is C.

QUESTION 1.29

Q 29: Imagine while preparing for the SCJP exam you created the following program to understand the concept of enum:

```
package com.kogent;
```

```
public class Ques66 {
        public enum Months { JAN, FEB, MARCH, APRIL, MAY };
        public static void main(String args[]) {
                for(Months m: Months.values());
                Months[] m1 = Months.values();
                System.out.println(m1[4]);
        }
}
```

What will be output of the preceding program?

A. The program will compile successfully and display "APRIL" as output.
B. The program will lead to compilation error.
C. The program will compile successfully but lead to runtime exception.
D. The program will compile successfully and display "MAY" as output.

A 29: The correct option is D.

Explanation: The values() method is invoked on the Months enum which returns an array of the enum values in the order of their declaration in an enum. Therefore, the preceding program displays the 4th element of Months array i.e MAY. As a result, the correct option is D.

QUESTION 1.30

Q 30: Which of the following names adhere to the JavaBeans standard?

A. getName()
B. addLength()
C. deleteId()
D. isAuthenticate()

A 30: The correct option is A and D.

Explanation: The A and D use valid prefixes get and is, as per the JavaBeans naming standards and so A and D are valid options.