Started on	Saturday, 29 February 2020, 4:33 PM
State	Finished
Completed on	Saturday, 29 February 2020, 4:38 PM
Time taken	4 mins 25 secs
Marks	7.33/8.00
Grade	**91.67** out of 100.00
Feedback	Congratulations!!! You have passed by securing more than 80%

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

___ and _____ are the access specifiers that can be applied to top level Class.

Select one or more:

- a. default ✔
- b. virtual
- c. public ✔
- d. protected

Your answer is correct.

The correct answers are: default, public

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

```
class Sample{
  private double num = 100;
  private int square(int a){
      return a*a;
  }
}
public class Test{
  public static void main(String args[]){
      Sample obj = new Sample();
      System.out.println(obj.num);
      System.out.println(obj.square(10));
  }
}
```

Quiz navigation

1 2 3 4 5 6 7 8

Show one page at a time

Finish review

Select one:

- a. Run time error
- b. 100
- c. Executes but no output
- d. Compile time error ✔

Your answer is correct.

The correct answer is: Compile time error

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

Choose the appropriate access specifier for the attribute value so that it can be accessed from anywhere.

```
class Test
{
public ✔ int value;
}
```

Your answer is correct.

The correct answer is:
Choose the appropriate access specifier for the attribute value so that it can be accessed from anywhere.

```
class Test
{
[public] int value;
}
```

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Choose the appropriate return type for the getters and setters provided below.

```
class Test
{
private int value;
public void ✔ setValue(int value){//some code}
public int ✔ getValue(){//some code}
}
```

Your answer is correct.

The correct answer is:
Choose the appropriate return type for the getters and setters provided below.

```
class Test
{
private int value;
public [void] setValue(int value){//some code}
public [int] getValue(){//some code}
}
```

Question 5

Partially correct

Mark 0.33 out of 1.00

Flag question

Analyze the below program, and fill the correct code so that it produces the below output:

```
0.0
101

public class Book {
        private int bookId;
        private double bookPrice;
        public int getBookId() {
                return bookId;
        }
        public void setBookId(int bookId) {
                this.bookId = bookId;
        }
        public double getBookPrice() {
                return bookPrice;
        }
        public void setBookPrice(double bookPrice) {
                this.bookPrice = bookPrice;
        }

}

class Test
{
        public static void main(String a[])
        {
                Book bobj=new Book();
```

- bobj.setBookId(101); ✗
- System.out.println(bobj.getBookPrice()); ✗
- System.out.println(bobj.getBookId()); ✓

```
        }
}
```

Your answer is partially correct.
Once we create an object, the default value will be assigned for each attribute. First print the values, then set the values and again print the values.

You have correctly selected 1.
The correct answer is: Analyze the below program, and fill the correct code so that it produces the below output:

```
0.0
101

public class Book {
        private int bookId;
        private double bookPrice;
        public int getBookId() {
                return bookId;
        }
        public void setBookId(int bookId) {
                this.bookId = bookId;
        }
        public double getBookPrice() {
                return bookPrice;
        }
        public void setBookPrice(double bookPrice) {
                this.bookPrice = bookPrice;
        }

}

class Test
{
        public static void main(String a[])
        {
                Book bobj=new Book();
                [System.out.println(bobj.getBookPrice());]
                [bobj.setBookId(101);]
                [System.out.println(bobj.getBookId());]


        }
}
```

Question 6

Correct

Mark 1.00 out of 1.00

Flag question

Consider the below code snippet and determine the output.

```
class Student
{    private int studentId;
     private float average;
}
class Test
```

```
{
    public static void main(String a[])
    {
        Student s=new Student();
        s.studentId=123;
        System.out.println(s.studentId);
    }
}
```

Select one:

- a. 0
- b. Any value
- c. Compile time error ✔
- d. 1

Your answer is correct.
Private variables can be accessed only within the class. They cannot be accessed outside the class.

The correct answer is: Compile time error

Question 7

Correct

Mark 1.00 out of 1.00

Flag question

The below code snippet shows an error

cannot find symbol:

System.out.println("BookId:"+bobj.getId());

```
public class Book {
    private int bookId;
    private double bookPrice;
    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public double getBookPrice() {
        return bookPrice;
    }
    public void setBookPrice(double bookPrice) {
        this.bookPrice = bookPrice;
    }
```

```
}
public class Test {
    public static void main(String[] args) {
        Book bobj=new Book();

        bobj.setBookId(123);
        bobj.setBookPrice(500);
        System.out.println("BookId:"+bobj.getId());
        System.out.println("BookPrice:"+bobj.getBookPrice());
    }
}
```

Analyze the above code and select the correct reason for the error.

Select one:

- a. getId method is not present in the book class ✔
- b. Getter method should not be called inside System.out.println
- c. "+" symbol should not be used in System.out.println
- d. bobj is not initialized

Your answer is correct.
When we specify the methodname or variablename or classname wrongly,then we will get an error "cannot find symbol". The compiler tries to fetch the methodname "getId" from the book class, where it is not defined.

The correct answer is: getId method is not present in the book class

Question 8

Correct

Mark 1.00 out of 1.00

Flag question

Arrange the code in the correct sequence, so that the program compiles successfully.

✔
```
public class Employee {
private int employeId;
private float salary;
```

✔
```
public void setSalary(float salary1) {
```

✔
```
if(salary>0){
salary=salary1;
}
```

✔
```
}
```

```
}
```

Your answer is correct.
The purpose of the setter method is to set a valid value for the attribute, by doing the necessary validations.

Finish review