Part I	Part II	Part III		Total
/10	/40	/50		/100

Part I TRUE/FALSE 10 Marks

For each question below, there is only one answer, write either TRUE or FALSE as your answer, not T or F.

1. Every Java source code file, such as “Prog.java”, produces one byte code file, such as “Prog.class”.

   Your Answer: FALSE

2. The constructor of a class cannot be private.

   Your Answer: FALSE

3. Every abstract class must have at least one abstract method.

   Your Answer: FALSE

4. In Java, every object must be created using the “new” operator.

   Your Answer: FALSE

5. Java provides a default constructor for each interface.

   Your Answer: FALSE

6. One can use a variable “obj” of “Object” type declared as “Object obj” to refer to any object of any class in a program.

   Your Answer: TRUE

7. The entry point of a Java program, namely, the “public static void main(String args[ ])” method, cannot throw any exception.

Your Answer: FALSE

8. One can always extend an existing class to produce a subclass.

Your Answer: FALSE

9. A class can be either abstract or final, but not both at the same time.

Your Answer: TRUE

10. After assigning a subclass reference variable to a superclass reference variable, this superclass reference variable can be used to invoke only the methods declared in the superclass.

Your Answer: TRUE

Part II Multiple Choices 40 Marks

Each question has only one correct answer *unless otherwise specified*. Any incorrect answer will result in a mark of zero for the question.

1. Consider the following code fragment:

```
1. class StudentProb {
2.     private int studentId = 0;
3.     void setStudentID(int sid) {
4.             studentId = sid;
5.             System.out.println("Student ID has been set to " + sid);
6.     }
7.     public static void main(String args[]) {
8.             int i = 420;
9.             Object ob1;
10.            StudentProb st1 = new StudentProb();
11.            ob1 = st1;
12.            st1.setStudentID(i);
13.    }
14. }
```

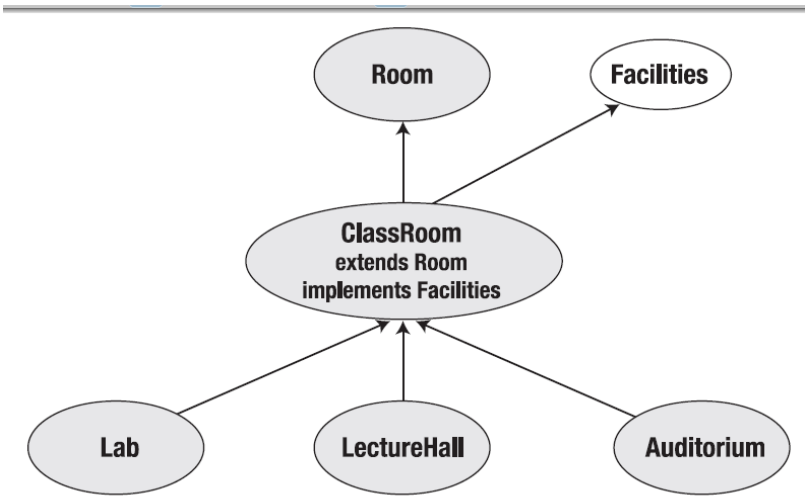
Which of the following statements is true about this code?

A. A compiler error will occur due to line 9.
B. A compiler error will occur due to line 11.
C. An exception will occur during execution due to line 11.

D. The code will compile, execute, and produce the output "Student ID has been set to 420."

Your Answer: D.

For Question 2 and 3, consider the class hierarchy show below:



2. Consider the following code fragment:

```
1. LectureHall lh = new LectureHall();
2. Auditorium a1;
3. Facilities f1;
4.
5. f1 = lh;
6. a1 = f1;
```

What of the following is the true statement about this code?

A. The code will compile and execute without any error.
B. Line 5 will generate a compiler error because an explicit conversion (cast) is required.
C. Line 6 will generate a compiler error because an explicit conversion (cast) is required to convert Facilities to Auditorium.

Your Answer: C

3. Consider the following code fragment:

```
1. LectureHall lh = new LectureHall();
2. Auditorium a1;
3. Facilities f1;
4.
5. f1 = lh;
6. a1 = (Auditorium) f1;
```

What of the following is the true statement about this code?

A. The code will compile and execute without any error.
B. Line 5 will generate a compiler error because an explicit conversion (cast) is required.
C. Line 6 will generate a compiler error because an interface cannot be converted to the class that implements the interface.
D. Line 6 will compile fine, but an exception will be thrown during the execution time.

Your Answer: D

4. Which of the following is true about an interface?

A. You can declare a method with the private modifier in an interface.
B. You must use the public modifier while declaring a method in an interface.
C. You can declare variables inside an interface, and change their values in a class that implements the interface.
D. You can declare variables inside an interface, but you cannot change their values in a class that implements the interface.
E. You cannot declare a variable inside an interface.

Your Answer: D

5. Consider the following code:

```
1. class MyClass {
2.     int i = 5;
3.     static int j = 7;
4.     public static void printSomething () {
5.         System.out.println("i: " + i);
6.         System.out.println("j: " + j);
7.     }
8. }
```

What is the result if the printSomething() method of MyClass is called from another class?

A. i: 5
   j: 7
B. A compiler error occurs at line 5.
C. A compiler error occurs at line 6.

D. A runtime exception is thrown.

Your Answer: B

6. Consider the following class definition:

```
public class SubClass extends SuperClass {
        public SubClass ( int i){
        }
        public SubClass(int i, int j) {
                super(i, j);
        }
        public SubCla ss (int i, int j, int k) {
        }
}
```

Which of the following forms of constructors must exist in the SuperClass? **(Multiple answers)**

A. SuperClass(int i) { }
B. SuperClass(int i, int j) { }
C. SuperClass() { }
D. SuperClass(int i, int j, int k) { }

Your Answer: B and C

7. An abstract method cannot be modified by:

A. public
B. protected
C. private
D. none of the above

Your Answer: C

8. A runtime exception is a/an:

A. checked exception
B. unchecked exception
C. offending exception
D. none of the above

Your Answer: B

9. The default layout manager for **JFrame** is:
   A. FlowLayout
   B. BorderLayout
   C. GridLayout
   D. None of the above.

Your Answer: B

10. Consider the following code:

```
1. class Foo {
2.       static boolean condition;
3.       public static void main(String [] args) {
4.              int i = 0;
5.              if((++i >= 1) || (condition == false))
6.                     i++;
7.              if((i++ > 1) && (condition = =true))
8.                     i++;
9.              System.out.println(i);
10.      }
11. }
```

What is the result of this code?
   A. 4
   B. 3
   C. 2
   D. 1
   E. Compiler error at line 7
   F. Throws exception at runtime

Your Answer: B

11. Consider the following code:

```
1. class Car extends Vehicle {
2.       public static void main(String [] args) {
3.                new Car().run();
4.        }
5.        private final void run() {
6.                System.out.println("Car");
7.       }
8. }
9. class Vehicle {
10.      private final void run() {
11.               System.out.println("Vehicle");
12.      }
13. }
```

What is the result?

A. Car
B. Vehicle
C. Compiler error at line 3
D. Compiler error at line 5
E. Exception thrown at runtime

Your Answer: A

12. Consider the following code:

```
1. class MySuperClass {
2.        private int x;
3.       MySuperClass(int i){
4.                x=i;
5.                System.out.println("mySuperClass: "+ x);
6.       }
7.        // Insert code here.
8. }
9. class MySubClass extends MySuperClass {
10.       public static void main(String[] args){
11.               new MySubClass();
12.               new MySubClass(3);
13.      }
14.      MySubClass(int i){
15.               super(i);
16.      }
17.      MySubClass() {
18.               // Insert code here
19.               System.out.println("Default");
20.      }
21. }
```

Which of the following two actions will make this code compile and execute?
1. Insert MySuperClass(){} at line 7.
2. Insert this(5); at line 18.

A. 1 only.
B. 2 only.
C. Either 1, or 2, or both.
D. Only 1 and 2.
E. Neither.
F. Either 1 or 2 will make the code compile but it will throw an exception at runtime.

Your Answer: C

13. Consider the following program “Abs.java”.

```
abstract class Base{
        public void myfunc() {
                System.out.println("Myfunc() in abstract class.");
        }
        public void another(){
                System.out.println("Another method");
        }
}
public class Abs extends Base{
        public static void main(String argv[]){
                Abs a = new Abs();
                a.amethod();
        }
        public void myfunc(){
                System.out.println("My Func in Abs class.");
        }
        public void amethod(){
                myfunc();
        }
}
```

Which of the following statements best describes the above program?
A. The code will compile and run, printing out the words "My Func in Abs class.".
B. The compiler will complain that the Base class has no abstract methods
C. The code will compile but complain at run time that the Base class has no abstract methods
D. The code will compile and run, printing out the words "MyFunc in abstract class.”

Your Answer: A

14. Consider the following method declared in a class whose header is below:

"public myMethod() throws AnException"

Which of the following statements is true?

A. There must have a try statement in the method "myMethod".
B. The exception "AnException" must be a checked exception.
C. The exception "AnException" was thrown by a throw statement in "myMethod".
D None of the above.

Your answer: D

15. Which statements, when inserted at the indicated position in the following code, will cause a runtime exception?

```
class A {}

class B extends A {}

class C extends A {}

public class Q3ae4 {
   public static void main(String[] args) {
      A x = new A();
      B y = new B();
      C z = new C();
      // insert statement here
   }
}
```

A. x=y;

B. z=x;

C. y=(B)x;

D. z=(C)y;

E. y=(A)y;

Your Answer: C

16. Consider the following progam:

```
class Vehicle {
        public void drive() {
                System.out.println("Vehicle: drive");
        }
        public String aVehicle(){
                return "A Vehicle.";
        }
}
class Car extends Vehicle {
        public void drive() {
                System.out.println("Car: drive");
        }
        public String aCar() {
                return "A Car.";
        }
}
public class Test_1 {
        public static void main (String args []) {
                Vehicle v;
                Car c;
                v = new Vehicle();
                c = new Car();
                v.drive();
                c.drive();
                v = c;
                v.drive();
                ((Vehicle)v).drive();
        }
}
```

A. Compile time error because of unhandled checked exception.

B. Runtime error.

C. Output is:
   Vehicle: drive
   Car: drive
   Car: drive
   Car: drive

D. None of the above.

Your Answer: C

17. Consider the following program.

```
public class UsingExceptions {
  public static void main( String args[] ) {
    try {
      throwException();
    }
    catch ( Exception exception )
    {
      System.out.println( "Exception handled in main" );
    }
    doesNotThrowException();
  }
  public static void throwException() throws Exception {
    try {
      System.out.println( "Method throwException" );
      throw new Exception();
    }
    catch ( Exception exception )
    {
      System.out.println(
        "Exception handled in method throwException" );
     }
    finally {
      System.out.println( "Finally executed in throwException" );
      throw new Exception();
    }
  }
  public static void doesNotThrowException(){
    try {
      System.out.println( "Method doesNotThrowException" );
    }
    catch( Exception exception )
    {
      System.out.println( exception );
    }
    finally {
      System.out.println(
        "Finally executed in doesNotThrowException" );
    }
    System.out.println( "End of method doesNotThrowException" );
  }
}
```

Which of the following statement is true?

A. Compile-time error because an exception can not be thrown from a finally block.

B. Compiles. But runtime error occurs because of uncaught exception.

C. Runs with the output:

```
Method throwException
Exception handled in method throwException
Finally executed in throwException
Exception handled in main
Method doesNotThrowException
Finally executed in doesNotThrowException
End of method doesNotThrowException
```

D. None of the above.

Your Answer: C

18. What is the output of the following program?

```
public class Inherit
{
   class Figure
   {
        void display( )
        {
             System.out.println("Figure");
        }
   }
   class Rectangle extends Figure
   {
        void display( )
        {
             System.out.println("Rectangle");
        }
   }
   class Box extends Rectangle
   {
        void display( )
        {
             System.out.println("Box");
        }
   }
   Inherit( )
   {
        Figure f = new Figure( );
        Rectangle r = new Rectangle( );
        Box b = new Box( );
        f.display( );
        f = r;
        ((Figure) f).display( );
        f = (Figure) b;
        f.display( );
   }

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

A. Figure Rectangle Box	B. Figure Figure Figure	C. Figure Rectangle Figure
Rectangle Figure	E. The program will not compile, therefore, no output.	

Your Answer: A

19. Suppose A is an interface, B is a concrete class that implements A with a default constructor. Which of the following is correct? **(Multiple answers)**

A. B b = new A();

B. A a = new A();

C. B b = new B();

D. A a = new B();

Your Answer: C, D

20. Given the following code, find the syntax error? **(Multiple answers)**

```
public class Test {
        public static void main(String[] args) {
                m(new GraduateStudent());
                m(new Student());
                m(new Person());
                m(new Object());
        }
        public static void m(Student x) {
                System.out.println(x.toString());
        }
}
class GraduateStudent extends Student {
}
class Student extends Person {
        public String toString() {
                return "Student";
        }
}
class Person extends Object {
        public String toString() {
                return "Person";
        }
}
```

A. m(new Person()) causes an error.

B. m(new Student()) causes an error.

C. m(new Object()) causes an error.

D. m(new GraduateStudent()) causes an error.

Your Answer: A, C

Part III Programming 50 marks

Question 1: （ 15 marks）

Consider the classes “MyClass”, “MyExtendedClass” and “MyFinalClass”.

```
public class MyClass{
        public int v1;
        public int v2;
        public MyClass(){
                v1 = 1;
                v2 = 2;
        }
        public void method1(){
                System.out.println("method1 in MyClass "+v1);
        }
        public void method2(){
                System.out.println("method2 in MyClass "+v2);
        }
        public void method3(){
                System.out.println("method3 in MyClass "+v2);
                this.method2();
        }
}
```

```
public class MyExtendedClass extends MyClass{
        public int v2;
        public MyExtendedClass(){
                v1=11;
                v2=22;
        }
        public void method2(){
                System.out.println("method2 in MyExtendedClass "+v2);
                this.method1();
        }
}
```

```
public final class MyFinalClass extends MyExtendedClass{
        public int v1;
        public MyFinalClass(){
                v1 = 111;
        }
        public void method1(){
                System.out.println("method1 in MyFinalClass "+v1);
                super.method1();
        }
}
```

These classes, as well as the class “Test” given below, do not contain errors.

What is the output produced after running the application Test?

```
public class Test{

  public static void main(String args[]){

        System.out.println("FIRST CASE");
        MyClass obj1 = new MyClass();
        obj1.method1();
        obj1.method2();
        obj1.method3();

        System.out.println("SECOND CASE");
        MyClass obj2 = new MyExtendedClass();
        obj2.method1();
        obj2.method2();
        obj2.method3();

        System.out.println("THIRD CASE");
        MyClass obj3 = new MyFinalClass();
        obj3.method1();
        obj3.method2();
        obj3.method3();

        System.out.println("FOURTH CASE");
        MyExtendedClass obj4 = new MyFinalClass();
        obj4.method2();

  }
}
```

Your Answer:

FIRST CASE
method1 in MyClass 1 ->**0.5**
method2 in MyClass 2 -> **0.5**
method3 in MyClass 2 ->**0.5**
method2 in MyClass 2 ->**0.5**
SECOND CASE
method1 in MyClass 11 -> **0.5 mark**
method2 in MyExtendedClass 22 -> **1 marks**
method1 in MyClass 11 -> **1 mark**
method3 in MyClass 2 -> **0.5 mark**
method2 in MyExtendedClass 22 -> **0.5 mark**
method1 in MyClass 11 -> **0.5 mark**
THIRD CASE
method1 in MyFinalClass 111 -> **1 marks**
method1 in MyClass 11 -> **1 marks**
method2 in MyExtendedClass 22 -> **0.5 mark**
method1 in MyFinalClass 111 -> **0.5 mark**
method1 in MyClass 11 -> **0.5 mark**
method3 in MyClass 2 -> **0.5 mark**
method2 in MyExtendedClass 22 -> **0.5 mark**
method1 in MyFinalClass 111 -> **0.5 mark**
method1 in MyClass 11 -> **1 mark**
FOURTH CASE
method2 in MyExtendedClass 22 -> **1 mark**
method1 in MyFinalClass 111 -> **1 mark**
method1 in MyClass 11 -> 1 marks

Question 2: （35 marks）

In this question, you will develop a class hierarchy of shapes and write a program that computes the amount of paint needed to paint different objects. **The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder.**

For the purposes of this program, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes).

Part of the class hierarchy is implemented in Java source code blow. You are not allowed to change the provided code. The code that you are going to implement should work together with the provided code below.

```
//  Sphere.java
public class Sphere extends Shape
{
    private double radius;  //radius in feet

   //--------------------------------
   //  Constructor: Sets up the sphere.
   //--------------------------------
   public Sphere(double r)
   {
      super("Sphere");
      radius = r;
   }

   //----------------------------------------
   //  Returns the surface area of the sphere.
   //----------------------------------------
   public double area()
   {
      return 4*Math.PI*radius*radius;
   }

   //--------------------------------
   //  Returns the sphere as a String.
   //--------------------------------
   public String toString()
   {
      return super.toString() + " of radius " + radius;
   }

}
```

```
//*******************************************************
//   Paint.java
//
//   Represents a type of paint that has a fixed area
//   covered by a gallon. All measurements are in feet.
//*******************************************************

public class Paint
{
   private double coverage;  //number of square feet per gallon

   //-----------------------------------------
   //  Constructor:  Sets up the paint object.
   //-----------------------------------------
   public Paint(double c)
   {
      coverage = c;
   }

   //-------------------------------------------------
   //  Returns the amount of paint (number of gallons)
   //  needed to paint the shape given as the parameter.
   //-------------------------------------------------
   public double amount(Shape s)
   {
      System.out.println ("Computing amount for " + s);
      return 0;
   }

}
```

```
//********************************************************
//  PaintThings.java
//
//  Computes the amount of paint needed to paint various
//  things. Uses the amount method of the paint class which
//  takes any Shape as a parameter.
//********************************************************

import java.text.DecimalFormat;

public class PaintThings
{
   //-----------------------------------------
   // Creates some shapes and a Paint object
   // and prints the amount of paint needed
   // to paint each shape.
   //-----------------------------------------
  public static void main (String[] args)
  {
     final double COVERAGE = 350;
     Paint paint = new Paint(COVERAGE);

     Rectangle deck;
     Sphere bigBall;
     Cylinder tank;

     double deckAmt, ballAmt, tankAmt;

     // Instantiate the three shapes to paint, for sub question 4(a)




     // Compute the amount of paint needed for each shape.
     // for sub question 4(b)


     // Print the amount of paint for each.
     DecimalFormat fmt = new DecimalFormat("0.#");
     System.out.println ("\nNumber of gallons of paint needed...");
     System.out.println ("Deck " + fmt.format(deckAmt));
     System.out.println ("Big Ball " + fmt.format(ballAmt));
     System.out.println ("Tank " + fmt.format(tankAmt));
  }
}
```

Do the following.

1. Write an abstract class **Shape** with the following properties: (**5 marks)**

An instance variable shapeName of type String **1 mark**
An abstract method area() **1 mark**
A toString method that returns the name of the shape **1 mark**
Any other method(s) needed. **Constructor 2 marks**

2. The file Sphere.java contains a class for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula 4*PI*radius^2. **Define similar classes for a rectangle and a cylinder.** Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is PI*radius^2*height. Define the toString method in a way similar to that for the Sphere class. **(18 marks)**

Rectangle class

Extends Shape 1 mark
Length variable 1 mark
Width variable 1 mark
Area method 2 marks
toString method 2 marks
Constructor 2 marks

Cylinder class

Extends Shape 1 mark
radius variable 1 mark
Height variable 1 mark
Area method 2 marks
toString method 2 marks
Constructor 2 marks

3. The file Paint.java contains a class for a type of paint (which has a "coverage" and a method to compute the amount of paint needed to paint a shape). **Correct the return statement in the amount method so the correct amount will be returned**. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint. (NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)

3 marks

4. The file PaintThings.java contains a program that computes the amount of paint needed to paint various shapes. A paint object has been instantiated. **Add the following to complete the program:**

a) Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30. **(6 marks)**

Each shape object instantiation **2 marks**

b) Make the appropriate method calls to assign the correct values to the three amount variables. **(3 marks)**

Each method call 1 mark