Question 1

Not yet answered

Marked out of 2.50

```
public class Hotel extends Travel {

    private double cost;

    public Hotel(double cost) {
        super() ;
    }

    public int getCost(){}

    public boolean equals(Object h) {}

    public String toString() {}
}
```

What Java principle is being demonstrated by the following lines of code?

public boolean equals(Object h) {}
public String toString() {}

Select one:

- A. Method Overriding
- B. Method Overloading
- C. Instance variables
- D. Parameter Passing

Question 2

Not yet answered

Marked out of 5.00

```
public class Hotel extends Travel {

    private double cost;

    public Hotel(double cost) {
        super() ;
    }

    public int getCost(){}

    public boolean equals(Object h) {}

    public String toString() {}
}
```

What is the name of the superclass in the code example above?

Select one:

- A. Hotel
- B. Class
- C. Travel
- D. Constructor

Question 3

Not yet answered

Marked out of 5.00

Given a double variable, double input = 3.1;, how would you cast the double variable to an integer?

Select one:

- A. (int) input;
- B. Double.parseDouble(input);
- C. Integer.toString(input);
- D. (double) input;

Question 4

Not yet answered

Marked out of 2.50

```
String type = "reptile";
int legs = 6;
boolean dangerous = true;
String pattern = "striped";

      if (type.equals("mammal")) {
         if (legs == 4) {
            if (dangerous == true) {
               if (pattern.equals("solid")) {
                  lion();
               } else {
                  tiger();
               }
            } else {
               cat();
            }
         } else {
            System.out.println("Not classified");
         }
      } else if (type.equals("reptile")) {
         if (legs == 4) {
            if (dangerous == true) {
               crocodile();
            } else {
               lizard();
            }
         } else if (legs == 0) {
            snake();
         } else {
            System.out.println("Not Classified");
         }

      } else if (type.equals("insect")) {
         if (legs == 8) {
            spider();
         } else if (legs == 6) {
            beetle();
         } else {
            System.out.println("Not Classified");
         }
      } else {
         System.out.println("Not Classified");
      }
```

Based on the code above, what is the correct description for the if statement given below?

```
if (type.equals("reptile")) {
   if (legs == 4) {
      if (dangerous == true) {
         crocodile();

} } }
```

Select one:

- A. If the type is equal to the value of “reptile”, and if the number of legs is equal to four, and if the boolean variable dangerous is not set, then call the crocodile method.
- B. If the type is equal to the value of “reptile”, and if the number of legs is equal to four, and if the boolean variable dangerous is set to true, then call the crocodile method.
- C. If the type is equal to the value of “reptile”, and if the number of legs is equal to six, and if the boolean variable dangerous is set to false, and if the pattern variable is set to “solid”, then do not call the crocodile method.
- D. If the type is equal to the value of “reptile”, and if the number of legs is equal to four, and if the boolean variable dangerous is set to false, then call the crocodile method.

Question 5

Not yet answered

Marked out of 5.00

```
public class Hotel extends Travel {

    private double cost;

    public Hotel(double cost) {
        super() ;
    }

    public int getCost(){}

    public boolean equals(Object h) {}

    public String toString() {}
}
```

What type of variable is the following code?

private double cost;

Select one:

- A. Instance variable
- B. Local variable
- C. Static variable
- D. Public variable

Question 6

Not yet answered

Marked out of 2.50

Given the String input below, how would we produce the output “Exam”?

String input = "CPT105 Exam Today";

Select one:

- A. input.substring(0,6);
- B. input.substring(6, 11);
- C. input.substring(7, 11);
- D. input.substring(7);

Question 7

Not yet answered

Marked out of 2.50

We wish to create a method named calculate that will receive an integer and a double as arguments and return an integer. Which method header is correct?

Select one:

- A. public static double calculate(double a,b){
- B. public static int calculate(int a, double b) {
- C. public static int calculate(int a, int b);
- D. public static void calculate(int a, double b){

Question 8

Not yet answered

Marked out of 50.00

You are required to fill 6 blocks to implement the following Java project. The project contains two classes: Bicycle and MountainBike, and MountainBike is a subclass of Bicycle:

Test Case:

```
public class Test {
    public static void main(String args[])
    {

        MountainBike mb = new MountainBike( gear: 3,  speed: 100,  startHeight: 25);
        System.out.println(mb.toString());
    }
}
```

```
// base class
class [Bicycle] {
    // the Bicycle class has two fields
    public int gear;
    public int speed;

    // the Bicycle class has one constructor
    public Bicycle(int gear, int speed)
    {
        [this.gear] = gear;
        [this.speed] = speed;
    }

    // the Bicycle class has three methods
    public void applyBrake(int decrement)
    {
        speed -= decrement;
    }

    public void speedUp(int increment)
    {
        speed += increment;
    }

    // toString() method to print info of Bicycle
    public String toString()
    {
        return ("No of gears are " + gear + "\n"
                + "speed of bicycle is " + speed);
```

```
    }
}

class MountainBike extends Bicycle {

    // the MountainBike subclass adds one more field
    public int seatHeight;

    // the MountainBike subclass has one constructor
    public MountainBike(int gear, int speed,
                        int startHeight)
    {
        super(gear, speed);
        seatHeight = startHeight;
    }

    // the MountainBike subclass adds one more method
    public void setHeight(int newValue)
    {
        seatHeight = newValue;
    }


    @Override public String toString()
    {
        return (super.toString() + "\nseat height is "
                + seatHeight);
    }
}
```

Question 9

Not yet answered

Marked out of 2.50

```
String type = "reptile";
int legs = 6;
boolean dangerous = true;
String pattern = "striped";

      if (type.equals("mammal")) {
         if (legs == 4) {
            if (dangerous == true) {
               if (pattern.equals("solid")) {
                  lion();
               } else {
                  tiger();
               }
            } else {
               cat();
            }
         } else {
            System.out.println("Not classified");
         }
      } else if (type.equals("reptile")) {
         if (legs == 4) {
            if (dangerous == true) {
               crocodile();
            } else {
               lizard();
            }
         } else if (legs == 0) {
            snake();
         } else {
            System.out.println("Not Classified");
         }

      } else if (type.equals("insect")) {
         if (legs == 8) {
            spider();
         } else if (legs == 6) {
            beetle();
         } else {
            System.out.println("Not Classified");
         }
      } else {
         System.out.println("Not Classified");
      }
```

When this code is run, what method is called?

Select one:

- A. The tiger() method is called.
- B. The beetle() method is called.
- C. No method is called, but “Not Classified” is displayed.
- D. The cat() method is called.

Question 10

Not yet answered

Marked out of 2.50

```
        [B]
private int getFullNum (int idNumber){
}
```

Looking at the code given here, what is represented by B?

Select one:

- A. Access modifier for the method
- B. Argument or parameter to the method
- C. Return value for the method
- D. Return type for the method

Question 11

Not yet answered

Marked out of 2.50

We wish to create a method named checkAndDisplay that will receive 2 boolean arguments, but not return anything. Which method header is correct?

Select one:

- A. public static boolean checkAndDisplay (boolean a) {
- B. public static void checkAndDisplay (boolean a, boolean b){
- C. public static void checkAndDisplay (boolean a, boolean b) ;
- D. public static boolean checkAndDisplay(String a, String b){

Question 12

Not yet answered

Marked out of 2.50

Given the String input below, how would we produce the output "TODAY"?

String input = "CPT105 Exam Today";

Select one:

- A. input = input.substring(7, 11).toUpperCase();
- B. input = input.substring(input.length()-5, input.length()).toUpperCase();
- C. input = input.toUpperCase (input.length()-5, input.length());
- D. input = input.toUpperCase (input.length()-5, input.length()).equalsIgnoreCase();

Question 13

Not yet answered

Marked out of 2.50

```
String type = "reptile";
int legs = 6;
boolean dangerous = true;
String pattern = "striped";

      if (type.equals("mammal")) {
         if (legs == 4) {
            if (dangerous == true) {
               if (pattern.equals("solid")) {
                  lion();
               } else {
                  tiger();
               }
            } else {
               cat();
            }
         } else {
            System.out.println("Not classified");
         }
      } else if (type.equals("reptile")) {
         if (legs == 4) {
            if (dangerous == true) {
               crocodile();
            } else {
               lizard();
            }
         } else if (legs == 0) {
            snake();
         } else {
            System.out.println("Not Classified");
         }

      } else if (type.equals("insect")) {
         if (legs == 8) {
            spider();
         } else if (legs == 6) {
            beetle();
         } else {
            System.out.println("Not Classified");
         }
      } else {
         System.out.println("Not Classified");
      }
```

Another way to write the if statements above is to use multiple condition if statements. If we keep the condition values the same, what code will correctly call the cat() method?

Select one:

A. if (type.equals("mammal") && legs == 4) {

B. if (type == "mammal" && legs == 4 && dangerous==true) {

C. if (type.equals("mammal") && legs == 4 && dangerous == false) {

D. if (type.equals("mammal") && legs == 4 && dangerous) {

Question 14

Not yet answered

Marked out of 2.50

```
public class Hotel extends Travel {

    private double cost;

    public Hotel(double cost) {
        super() ;
    }

    public int getCost(){}

    public boolean equals(Object h) {}

    public String toString() {}
}
```

What does the following line of code do?

super();

Select one:

A. A new Hotel object is created

B. The constructor of the Travel object is called

C. The constructor of the Hotel object is called

D. The variables are reset

Question 15

Not yet answered

Marked out of 2.50

```
double total = 100;
while (total<5){
   total = total -10;
 }
System.out.println(total);
```

This while loop will run 0 times. Why?

Select one:

A. The double variable is not declared.

B. The use of curly brackets is not correct.

C. The condition should be <= 5.

D. The condition is never true.

Question 16

Not yet answered

Marked out of 2.50

```
public class Hotel extends Travel {

    private double cost;

    public Hotel(double cost) {
        super() ;
    }

    public int getCost(){}

    public boolean equals(Object h) {}

    public String toString() {}
}
```

What Java principle does the following code demonstrate:

private double cost;
public int getCost(){}

Select one:

- A. Method overloading
- B. Inheritance
- C. Encapsulation
- D. Shared variables

Question 17

Not yet answered

Marked out of 2.50

What Java Class would you use to calculate the difference between dates?

Select one:

- A. The Period Class
- B. The Date Class
- C. The Duration Class
- D. The getDate() Class

Question 18

Not yet answered

Marked out of 2.50

```
for(int i = 5; i > 1; i --){

  System.out.println("Value is " + i);
  counterDisplay(i);

}
```

How many times will this for loop run?

Select one:

- A. 5 times.
- B. 4 times.
- C. 1 time.
- D. Will not run.