BRAC University
Department of Computer Science and Engineering (CSE)

CSE111: Programming Language II

Quiz 4 – Spring 2025
Time: **25 minutes**
Full marks: **10**

A

Name: ____ ID: ____________________________ Section: _____

*(There is **1** question only. You must answer the complete java code.)*

You are designing a new inventory management system for a food retail company. Here, the given **Product** class is the parent class of the **Snacks** class.

Also, the **Chocolate** class inherits the **Snacks** class. For pricing, the company follows these rules:

For all **Snacks**:

1. If the package weight is 300 grams or more, then the price gets added with: **(4 * weight)** tk.
2. Otherwise, if the package weight is 100 grams or more, then the price is added with: **(6 * weight)** tk.
3. In case of all other snacks, the price gets an additional amount of: **(8 * weight)** tk.

For all **Chocolate**:

1. If the box has 30 or more pieces, then the price is discounted by: **(10 * quantity)** tk.
2. Otherwise, if the box has 15 or more pieces, then the price is reduced by: **(8 * quantity)** tk.
3. For smaller quantity, the chocolate's price gets a discount of: **(6 * quantity)** tk.

Now, write the **Snacks** and **Chocolate** classes such that the following **Tester** code generates the given output:

```java
public class Product {
public static int count;
public String name, id;
private double basePrice;

public Product(String name) {
  this.name = name;
  this.id = "#0" + (++count);
  this.basePrice = 100;
}

public double getPrice() {
  return basePrice;
}

public String toString() {
  String s1 = "Product: " + name;
  s1 += ", ID: " + id + "\nPrice: ";
  return s1 + getPrice()+" Tk";
}
}
```

Tester Code

```java
public class Tester {
public static void main(String[] args) {
  Snacks s1 = new Snacks("Noodles", 240);
  Snacks s2 = new Snacks("Cookies", 160);
  System.out.println(s1);
  System.out.println(s2);
  System.out.println("==========================");
  s1.addPackage(200);
  System.out.println("New Price: " + s1.getPrice());
  System.out.println("==========================");
  Chocolate c = new Chocolate("KitKat", 6, 75);
  System.out.println(c);
  System.out.println("==========================");
  c.quantity += 12;
  System.out.println(c);
  System.out.println("==========================");
  c.addPackage(50);
  System.out.println("New Price: " + c.getPrice());
  System.out.println("==========================");
  System.out.println(c);
}
}
```

Expected output

```
Product: Noodles, ID: S#01
Price: 1540.0 Tk
Weight: 240 grams
Product: Cookies, ID: S#02
Price: 1060.0 Tk
Weight: 160 grams
==========================
Noodles(ID: S#01) package now
weighs 440 grams.
New Price: 1860.0
==========================
New chocolate box created with 6
pieces.
Product: KitKat, ID: S#03C
Price: 664.0 Tk
Weight: 75 grams
Type: Chocolate, 6 pcs
==========================
Product: KitKat, ID: S#03C
Price: 556.0 Tk
Weight: 75 grams
Type: Chocolate, 18 pcs
==========================
KitKat(ID: S#03C) package now
weighs 125 grams.
New Price: 706.0
==========================
Product: KitKat, ID: S#03C
Price: 706.0 Tk
Weight: 125 grams
Type: Chocolate, 18 pcs
```

BRAC University
Department of Computer Science and Engineering (CSE)

CSE111: Programming Language II

Quiz 4 – Spring 2025
Time: **25 minutes**
Full marks: **10**

B

Name: ________________________________ ID: ______________________ Section: _____

*(There is **1** question only. You must answer the complete java code.)*

You are designing a new employee management system in a company. Here, the given `Employee` class is the parent class of the `FullTimeEmployee` class.

Also, the `Manager` class inherits the `FullTimeEmployee` class. For salary calculation, the company follows these rules:

For all `FullTimeEmployee`:

1. If experience is 10 years or more, then the salary gets added with: **`(8000 * experience)`** tk.
2. Otherwise, if experience is 5 years or more, then the salary is added with: **`(5000 * experience)`** tk.
3. For everyone else, the salary gets an additional amount of: **`(3000 * experience)`** tk.

For all `Manager`:

1. If their team has 30 or more people, then their salary is added with: **`(1200 * teamSize)`** tk.
2. Otherwise, if the team has 15 or more people, then their salary is added with: **`(800 * teamSize)`** tk.
3. For smaller teams, the manager's salary gets an additional amount of: **`(500 * teamSize)`** tk.

Now, write the `FullTimeEmployee` and `Manager` classes such that the following `Tester` code generates the given output:

```java
public class Employee {
public static int count;
public String name, id;
private double baseSalary;

public Employee(String name) {
  this.name = name;
  this.id = "#0" + (++count);
  this.baseSalary = 30000;
}

public double getSalary() {
  return baseSalary;
}

public String toString() {
  String s1 = "Name: " + name;
  s1 += ", ID: " + id + "\nSalary: ";
  return s1 + getSalary()+" Tk";
}
}
```

Tester Code

```java
public class Tester {
public static void main(String[] args) {
  FullTimeEmployee e1 = new FullTimeEmployee("Samia", 2);
  FullTimeEmployee e2 = new FullTimeEmployee("Rafid", 5);
  System.out.println(e1);
  System.out.println(e2);
  System.out.println("==========================");
  e1.addExperience(6);
  System.out.println("New Salary: " + e1.getSalary());
  System.out.println("==========================");
  Manager m = new Manager("Taosif", 21, 7);
  System.out.println(m);
  System.out.println("==========================");
  m.teamSize += 10;
  System.out.println(m);
  System.out.println("==========================");
  m.addExperience(5);
  System.out.println("New Salary: " + m.getSalary());
  System.out.println("==========================");
  System.out.println(m);
}
}
```

Expected output

```
Name: Samia, ID: Emp#01
Salary: 36000.0 Tk
Experience: 2 years
Name: Rafid, ID: Emp#02
Salary: 55000.0 Tk
Experience: 5 years
==========================
Samia(ID: Emp#01) now has 8
years' experience.
New Salary: 70000.0
==========================
New Manager role created with 21
team members.
Name: Taosif, ID: Emp#03M
Salary: 81800.0 Tk
Experience: 7 years
Role: Manager, Team size: 21
==========================
Name: Taosif, ID: Emp#03M
Salary: 102200.0 Tk
Experience: 7 years
Role: Manager, Team size: 31
==========================
Taosif(ID: Emp#03M) now has 12
years' experience.
New Salary: 163200.0
==========================
Name: Taosif, ID: Emp#03M
Salary: 163200.0 Tk
Experience: 12 years
Role: Manager, Team size: 31
```

Set A Solve

```java
class Snacks extends Product {
  int weight;

  public Snacks(String name, int wt) {
    super(name);
    weight = wt;
    id = "S" + id;
  }

  void addPackage(int y) {
    weight += y;
    System.out.println(name + "(ID: " + id + ")
package now weighs " + weight + " grams.");
  }

  public double getPrice() {
    if (weight >= 300) {
      return super.getPrice() + 4 * weight;
    } else if (weight >= 100) {
      return super.getPrice() + 6 * weight;
    } else {
      return super.getPrice() + 8 * weight;
    }
  }

  public String toString() {
    String s1 = super.toString();
    return s1 + "\nWeight: " + weight + " grams";
  }
}
```

```java
class Chocolate extends Snacks {
  int quantity;

  public Chocolate(String name, int qty, int wt) {
    super(name, wt);
    this.quantity = qty;
    id = id + "C";
    System.out.println("New chocolate box created
with " + qty + " pieces.");
  }

  public double getPrice() {
    if (quantity > 29) {
      return super.getPrice() - 10 * quantity;
    } else if (quantity > 14) {
      return super.getPrice() - 8 * quantity;
    } else {
      return super.getPrice() - 6 * quantity;
    }
  }

  public String toString() {
    String s1 = super.toString();
    return s1 + "\nType: Chocolate, " + quantity +
" pcs";
  }

}
```

Set B Solve

```java
class FullTimeEmployee extends Employee {
  int experience;

  public FullTimeEmployee(String name, int exp) {
    super(name);
    experience = exp;
    id = "Emp" + id;
  }

  void addExperience(int y) {
    experience += y;
    System.out.println(name + "(ID: " + id + ")
now has " + experience + " years' experience.");
  }

  public double getSalary() {
    if (experience > 9) {
      return super.getSalary() + 8000 *
experience;
    } else if (experience > 4) {
      return super.getSalary() + 5000 *
experience;
    } else {
      return super.getSalary() + 3000 *
experience;
    }
  }

  public String toString() {
    String s1 = super.toString();
    return s1 + "\nExperience: " + experience + "
years";
  }
}
```

```java
class Manager extends FullTimeEmployee {
  int teamSize;

  public Manager(String name, int team, int exp) {
    super(name, exp);
    this.teamSize = team;
    id = id + "M";
    System.out.println("New Manager role created
with " + team + " team members.");
  }

  public double getSalary() {
    if (teamSize > 29) {
      return super.getSalary() + 1200 * teamSize;
    } else if (teamSize > 14) {
      return super.getSalary() + 800 * teamSize;
    } else {
      return super.getSalary() + 500 * teamSize;
    }
  }

  public String toString() {
    String s1 = super.toString();
    return s1 + "\nRole: Manager, Team size: " +
teamSize;
  }

}
```