රුහුණ විශ්වවිද්‍යාලය
විද්‍යාවේදී (සාමාන්‍ය) උපාධිය
දෙවන ස්ථලය (පළමු සමාසිකය) විභාගය - අගෝස්තු 2017

COM212β – Object Oriented System Development

කාලය: පැය 2 යි

ප්‍රශ්න හතරකට පමණක් පිළිතුරු සපයන්න

1)

a) **Driver class** එකක් යනු කුමක්ද?

b) Class වල භාවිතා වන attribute වර්ග දෙක **(2)** නම් කරන්න. ඒවා අතර ප්‍රධාන වෙනස්කම විස්තර කරන්න.

c)

i. **Access Modifiers** වල කාර්යභාරය කෙටියෙන් විස්තර කරන්න.

ii. **Java Programming** වල යෙදෙන **public** සහ **protected** යන modifiers වල වෙනස accessibility දෘෂ්ටි කෝණයට අනුබද්ධව විස්තර කරන්න.

iii. පහත දී ඇති **Person Class** එක සහ **PersonDrive Class** එක සලකන්න.

Person Class

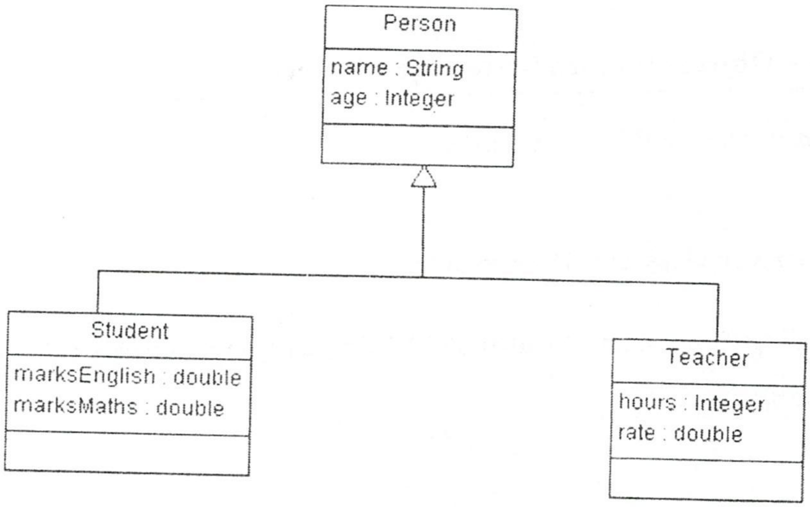
```
class Person
{
        public String name;
        private int age;
}
```

PersonDrive Class

```
class PersonDrive
{
        pubic static void main(String[] args)
        {
                Person bob=new Person();
                bob.name="Bob"; ← (A)
                System.out.println(bob.age);← (B)
        }
}
```

PersonDrive class එකෙහි **(A)** සහ **(B)** ප්‍රකාශ වලංගුද නැද්ද යන වග සඳහන් කරන්න. ඔබගේ පිළිතුර පහදන්න.

d) පහත දී ඇති class diagram කොටස සලකන්න.



i. **Student** සහ **Teacher** යන class එකිනෙක **Person** class එක සමග සම්බන්ධ කිරීමට භාවිතා කර ඇති **Object Oriented Design** සංකල්පය කුමක්ද? ඔබ සඳහන් කල සංකල්පය භාවිතා කිරීමේ වාසියක් සඳහන් කරන්න.

ii. ඔබ ඉහත **(i)** හි සඳහන් කළ සංකල්පය **Java Programming** වල යෙදෙන ආකාරය සඳහන් කරන්න.

iii. ඉහත class diagram එකෙහි දෙන ලද සැලසුම implement කිරීමට අවශ්‍ය **Java code** කොටස ලියන්න (access modifiers නොසලකා හරින්න).

2)

a) Rational Unified Process (RUP) හි best practices හතර (04) ලියා දක්වන්න.

b) පහත සඳහන් **Object Oriented Design Principles** කෙටියෙන් විස්තර කරන්න.

i. Abstraction
ii. Polymorphism

c)

i. **Java Programming** වල **constructor** එකක කාර්යභාරය කුමක්ද?

ii. Class එකක **default constructor** යන්නෙන් අදහස් වන්නේ කුමක්ද?

iii. පහත සඳහන් code කොටස සලකන්න.

```
class Course
{
        String course_id;
        String name;
        double credits;
}
```

ඉහත class එකෙහි එක් එක් attribute එක සඳහා අගයන් පැවරීමට **parameterized constructor** එක **Java** වලින් ලියන්න.

d)

i. Non-abstract classes වලට සාපේක්ෂව **abstract classes** වල ප්‍රධානතම වෙනස සඳහන් කරන්න.

ii. **Method Overriding** යන්නෙන් අදහස් වන්නේ කවරක්ද යන්න කෙටියෙන් විස්තර කරන්න.

iii. **String** අගයක් return කරන **displayAlert()** නම් වූ abstract method එකක් අඩංගු වන පරිදි **AbsMessage** නම් වූ abstract class එකක් සෑදීමට අවශ්‍ය **Java** code කොටස ලියන්න.

iv. ඔබ ඉහත **(iii)** හි ලියන ලද abstract class එකෙන් inherit කරන **MessageBuilder** නම් වූ class එකක් ලියන්න. "Message Builder Alert" යන message එක return වන පරිදි **displayAlert()** method එක override කරන්න.

3)

a) Rational Unified Process එකෙහි analysis phase එකේදී භාවිතා වන **UML** diagrams වර්ග දෙකක් **(02)** කෙටියෙන් විස්තර කරන්න.

b) Class diagram එකක් ඇඳීමේදී අනුගමනය කල යුතු සුදුසු ක්‍රියා පටිපාටියේ පියවර ලියන්න.

c)

i. Classes අතර පවතින **Composition** relationship එක කෙටියෙන් විස්තර කරන්න. Class diagram එකක් භාවිතයෙන් සුදුසු උදාහරණයක් දෙන්න.

ii. පහත විස්තරය සලකන්න

research project එකක් අවම වශයෙන් විශ්වවිද්‍යාලයේ එක department එකකින් හෝ මෙහෙයවනු ලැබේ. නමුත් දෙනලද research project එකක් උපරිම වශයෙන් department දෙකකට මෙහෙයවිය හැකිය. එසේම department එකකට research projects ශූන්‍ය සංඛ්‍යාවක් හෝ ඊට වැඩි සංඛ්‍යාවක් මෙහෙයවිය හැකිය.

ඉහත scenario එක සඳහා UML class diagram එකක් අඳින්න.

d) වාහන නිෂ්පාදන ආයතනයක කටයුතු හැසිරවීම සඳහා මෘදුකාංගයක් නිපදවීමට පහත සඳහන් කරුණු එක්රැස් කරගන්නා ලදී. දෙන ලද විස්තර සලකා UML class diagram එකක් අඳින්න.

- vehicle එකක් යනු vehicle parts වල එකතුවකි.
- vehicle parts වෙන වෙනම නිපදවීම සිදු කරන අතර ඒවා vehicle එකක් එකලස් කිරීමට යොදා ගැනේ.
- vehicle එකක vehicle parts ප්‍රතිස්ථාපනය කල හැකිය.
- Cars සහ Lorries මෙම ආයතනයේ නිෂ්පාදනය කෙරේ.
- එකලස් කල vehicle එකක් employee කෙනෙකු අතින් පරීක්ෂා කෙරේ.
- එක් employee කෙනෙකුට vehicles එකක් හෝ ඊට වැඩි ගණනක් පරීක්ෂා කල හැකිය.

4)

a) **Use Case Analysis** වල ප්‍රධාන අරමුණ කුමක්ද?

b) Software Development සඳහා **Use Case Analysis** භාවිතයේ වාසි හතරක් **(4)** ලියන්න.

c)

i. Use case diagrams වල use cases අතර වන **extends** relationship එකෙහි භාවිතය කෙටියෙන් විස්තර කරන්න.

ii. Use cases අතර පවතින **include** relationship එකෙහි වාසියක් සඳහන් කරන්න.

d) Online railway reservation system එකකට අදාළ පහත සඳහන් scenario එක සලකන්න. දී ඇති විස්තර ඇසුරින් **Use Case diagram** එකක් අඳින්න.

Online railway reservation system එක මගින් පාරිභෝගිකයන් සඳහා පහසුකම් කීපයක් සපයනු ලැබේ. පාරිභෝගිකයන්හට System එක භාවිතාකර trains තිබේද නොතිබේද යනවග සොයාබැලිය හැක. තවද, සෙවීමෙන් අනතුරුව පාරිභෝගිකයන්හට reservation request එකක් කල හැකිය. Reservation request එකක් කිරීමට ප්‍රථමයෙන් පද්ධතිය මගින් පාරිභෝගිකයන්ගේ අනන්‍යතාවය තහවුරු කරගත යුතුය.

Station එකෙහි සිටින clerk හට system එකට log වී පාරිභෝගිකයන්ගේ request වලට අනුව ආසන වෙන්කළ හැකිය. ආසන නොමැති නම් හෝ පාරිභෝගිකයා ගමනට අදාළ තොරතුරු හරිහැටි සඳහන් කර නොමැතිනම් reservation request එකක් ප්‍රතික්ෂේප කිරීමේ හැකියාව ඇයට තිබේ. Reservation එක සිදුකල පසු පාරිභෝගිකයා වෙත invoice එකක් යැවීම සිදුකෙරේ. පාරිභෝගිකයා ගෙවීම සිදුකල පසු, clerk ගෙවීම සලකා බලා reservation එක තහවුරු කරයි.

5)

a) **UML Sequence Diagrams** model එකක් යනු කුමක්ද?

b)

i. Sequence Diagrams වල වන **Lifelines** සහ **Activations** කෙටියෙන් විස්තර කරන්න.

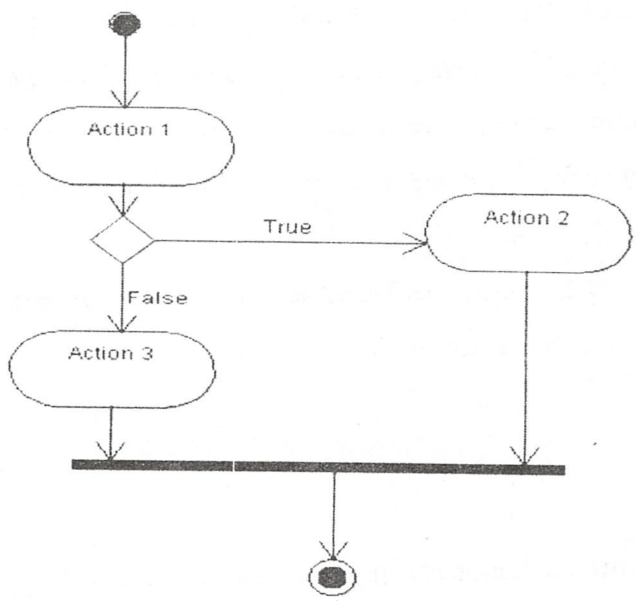
ii. Sequence Diagrams වල **Synchronous Messages** සහ **Asynchronous Messages** වල වෙනස විස්තර කරන්න.

c) Public key එකකට digital certificate එකක් ලබාගැනීම සම්බන්ධ පහත දැක්වෙන scenario එක සලකන්න.

මෙම ක්‍රියාවලියේදී, ප්‍රථමයෙන් web user public key එකක් ජනනය කල යුතුය. ඉන්පසු එය අනන්‍යතාව තහවුරු කිරීමේ අධිකාරියට යවනු ලැබේ. Public key එක ලැබුණු පසු, අනන්‍යතාව තහවුරු කිරීමේ අධිකාරියෙන් web user ගේ අනන්‍යතාව පරීක්ෂා කරනු ලැබේ. User ගේ අනන්‍යතාව තහවුරු වුවහොත් web user ගේ public key එක සඳහා certificate එකක් නිපදවේ. ඒ අතරම, අනන්‍යතාව සහතික කිරීමේ අධිකාරිය විසින් ඔවුන්ගේ logs යාවත්කාලීන කෙරේ. අවසානයේදී, public key certificate එක web user ට යවනු ලැබේ. User ගේ අනන්‍යතාව තහවුරු නොවුනහොත් failure notification එකක් user හට යවනු ලැබේ.

ඉහත scenario එක සඳහා swim lanes (partitions) දක්වමින් UML Activity diagram එකක් අඳින්න.

d) පහත සඳහන් Activity diagram එක සලකන්න.



i. ඉහත Activity diagram එකට අදාළ ගැටළුව විස්තර කරන්න.

ii. ඉහත (i) හි සඳහන් කල ගැටළුව විසඳීමට කිරීමට විසඳුමක් යෝජනා කරන්න.