



ASSIGNMENT COVER PAGE



Programme		Course Code and Title	
Bachelor of Computer Science (Hons)/ Bachelor of Computer Science (Hons) In Computer & Network Technology/ Bachelor of Software Engineering (Hons)		CPR3113/N Principles of Programming	
Student's name / student's id		Lecturer's name	
CHAN SEOW FEN / 0207368		Tan Phit Huan	
Date issued	Submission Deadline	Indicative Weighting	
Week 3 - 26/09/2022	Week 7 – 28/10/2022	30%	
Assignment 1 title		Selection and Iteration	

This assessment assesses the following course learning outcomes

# as in Course Guide	UOWM KDU Penang University College Learning Outcome
LO1	Analyze algorithms to solve basic computing problems using flow charts and pseudocodes.
LO2	Demonstrate a computational solution using principle of selection and iteration.
# as in Course Guide	University of Lincoln Learning Outcome
LO3	Demonstrate the ability to select from a range of possible options, to provide justification for that selection, and to implement algorithms in a particular context
LO1	Characterise a problem in the context of possible solution mechanisms
LO2	Model a problem solution using appropriate vocabulary
LO1	Implement control flow with decisions and loops using good programming practices
LO2	Determine an appropriate algorithmic approach to a problem
LO2	Implement control flow with decisions and loops using good programming practices
LO3	Apply object-oriented principles to the implementation of software programs

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature:

Chan

Submission Date:












23/10/2022

Table of Content

1.	Question 1	
	1.1 Defining diagram	3 - 4
	1.2 Pseudocode	4 - 7
	1.3 Desk checking table	7 - 10
2.	Question 2	
	2.1 Java program code	10 - 14
	2.2 Description	15 - 21

1.0 Question 1

1.1 Defining diagram

Input	Processing	Output
name identityCardNo s_height s_weight bloodType decision 	<p>Get patient's details (name, identityCardNo, height, weight, bloodType)</p> <p>Validate name input and repeat prompting process for any invalid input by checking it should only contains space and alphabet but not blank </p> <p>Validate identityCardNo input and repeat prompting process for any invalid input by checking it should contains 12 digits and it is numeric </p> <p>Get dob from the first six digits of identityCardNo and store them in DD/MM/YY date format </p> <p>Check if the last digit of identityCardNo is an odd number or an even number, then store gender as male or female </p> <p>Validate s_height input and repeat prompting process for any invalid input by checking it should be numeric and should not be zero or lesser than zero and higher than 3m </p> <p>Validate s_weight input and repeat prompting process for any invalid input by checking it should be numeric and should not be zero or lesser than zero and heavier than 640kg </p> <p>Validate bloodType input and repeat prompting process for any invalid input by checking if it is either A+, A-, B+, B-, O+, O-, AB+, or AB- </p> <p>Calculate bmi by using $\text{weight (kg)} / [\text{height(m)}]^2$</p> <p>Check if bmi is 30.0 and above or between 29.9 and 25.0 or between 24.9 and 18.5 or below 18.5, then store weightStatus as Obese or Overweight or Healthy Weight or Underweight </p> <p>Display the patient's information (name, identityCardNo, height, weight, bloodType, dob, gender, bmi, weightStatus) </p>	name identityCardNo height weight bloodType dob gender bmi weightStatus 

	<p>Get the administrator's decision to decide whether have a loop and continue to ask for the patient's details or end the program</p> <p>Validate decision input and repeat prompting process for any invalid input by checking if it is either yes or no</p>	
--	--	--

1.2 Pseudocode

START PatientInfo

```

1 INITIALISE validName to false
2 INITIALISE validIC to false
3 INITIALISE gender=""
4 INITIALISE height to 0
5 INITIALISE weight to 0
6 INITIALISE heightNumeric to true
7 INITIALISE weightNumeric to true

8  DOWHILE decision is 'Y' or 'y'

9      DOWHILE validName is false
10         GET name
11         IF name is blank THEN
12             DISPLAY "Invalid name, it should not be blank. Please input again."
13             validName=false
14         ELSE
15             FOR all the characters in the name
16                 IF character is NOT letter AND is NOT whitespace THEN
17                     DISPLAY "Invalid name, it should not contain special
18                     characters or numbers. Please input again."
19                     validName=false
20                     BREAK
21                 ELSE
22                     validName=true
23             ENDFOR
24         ENDDO
25     ENDDO

12  DOWHILE validIC is false
13     GET identityCardNo
14     IF length of identityCardNo is NOT equal to 12 THEN
15         DISPLAY "Invalid IC, it should contain 12 digits. Please input again."
16         validIC=false
17     ELSE
18         FOR all the characters in identityCardNo
19             IF character is NOT numeric THEN
20                 DISPLAY "Invalid IC, it should contain only numbers.
21                 Please input again."

```

```

                                validIC=false
                                BREAK
                                ELSE
                                validIC=true
                                ENDIF
                            ENDFOR
                        ENDIF
                    ENDDO

15    dob = fifth and sixth character from identityCardNo + "/" + third and fourth character from
    identityCardNo + "/" + first and second character from identityCardNo
16    IF last character of identityCardNo % 2 is NOT 0 THEN
        gender = Male
    ELSE
        gender = Female
    ENDIF

17    DOWHILE validHeight is false
18        GET s_height
19        TRY
            height = convert s_height from string datatype to double datatype
        CATCH execption THEN
            heightNumeric=false
        ENDTRY
20    IF heightNumeric is true THEN
        IF height>0 AND height<=3 THEN
            validHeight=true
        ELSE
            DISPLAY "Invalid height, it should not be negative value or higher
            than 3m. Please input again."
            validHeight=false
        ENDIF
    ELSE
        DISPLAY "Invalid height, please enter a valid height which is a number."
        validHeight=false
        heightNumeric=true
    ENDIF
    ENDDO

21    DOWHILE validWeight is false
22        GET s_weight
23        TRY
            weight = convert s_weight from string datatype to double datatype
        CATCH execption THEN
            weightNumeric=false
        ENDTRY
24    IF weightNumeric is true THEN
        IF weight>0 AND weight<=640 THEN
            validWeight=true
        ELSE

```

```

        DISPLAY "Invalid weight, it should not be 0 or lower than 0 or
        heavier than 640kg. Please input again."
        validWeight=false
    ENDIF
ELSE
    DISPLAY "Invalid weight, please enter a valid weight which is a number."
    validWeight=false
    weightNumeric=true
ENDIF
ENDDO

25 DOWHILE validBloodType is false
26     GET bloodType
27     IF bloodType equals to "A+" OR bloodType equals to "A-" OR bloodType equals
        to "B+" OR bloodType equals to "B-" OR bloodType equals to "O+" OR bloodType
        equals to "O-" OR bloodType equals to "AB+" OR bloodType equals to "AB-"
        THEN
            validBloodType=true
        ELSE
            DISPLAY "Invalid blood type. Please input again."
            validBloodType=false
        ENDIF
    ENDDO

28 bmi=weight/(height)2
29 IF bmi>=30 THEN
    weightStatus = "Obese"
ELSE IF bmi<30 AND bmi>=25 THEN
    weightStatus = "Overweight"
ELSE IF bmi<25 AND bmi>=18.5 THEN
    weightStatus = "Healthy Weight"
ELSE
    weightStatus = "Underweight"
ENDIF

30 DISPLAY "Name: "+name
31 DISPLAY "Identity card number: "+identityCardNo
32 DISPLAY "Height: "+height with two decimal place + ' m'
33 DISPLAY "Weight: "+weight with one decimal place + ' kg'
34 DISPLAY "Blood type: "+bloodType
35 DISPLAY "Date of birth[DD/MM/YY]: "+dob
36 DISPLAY "Gender: "+gender
37 DISPLAY "BMI: "+bmi with one decimal place
38 DISPLAY "Weight status: "+weightStatus

39 DOWHILE validDecision is false
40     GET decision
41     IF decision is NOT 'Y' AND decision is NOT 'y' AND decision is NOT 'N' AND
        decision is NOT 'n' THEN
            DISPLAY "Invalid input, please key in either Y or N."
            validDecision=false
        ELSE
```

```

        validDecision=true
    ENDIF
ENDDO
ENDDO

```

END PatientInfo

1.3 Desk checking table

Input data

	First data set	Second data set	
name	Tan Yu Sheng	Noor Arzila	,45489,Ali
identityCardNo	010528020721	750326070614	bdbc,020708070433
s_height	1.76	1.55	1.28.9,1.8
s_weight	80	48	1000,ad,100
bloodType	O+	AB-	dads,A+
decision	n	y	f,N

Expected output

	First data set	Second data set	
name	Tan Yu Sheng	Noor Arzila	Ali
identityCardNo	010528020721	750326070614	020708070433
height + ' m'	1.76 m	1.55 m	1.80 m
weight + ' kg'	80.0 kg	48.0 kg	100.0 kg
bloodType	O+	AB-	A+
dob	28/05/01	26/03/75	08/07/02
gender	Male	Female	Male
bmi	25.8	20.0	30.9
weightStatus	Overweight	Healthy Weight	Obese

Desk check table – First data set

	valid Name	valid IC	gender	s_ height	height	s_ weight	weight	height Numeric	weight Numeric	DO WHI LE cond ition	name	identity CardNo	dob	blood Type	decision	bmi	weight Status	valid Height	valid Weight	valid Blood Type	valid Decision
1,2,3,4,5,6,7	false	false			0		0	true	true												
8										false											
9										true											
10											Tan Yu Sheng										
11	true																				
9										false											
12										true											
13												010528020721									
14		true																			
12										false											
15													28/05/01								
16			Male																		
17										false											
18				1.76																	
19					1.76																
20																		true			
17										false											
21										false											
22						80															
23							80.0														
24																			true		
21										false											
25										false											
26														O+							
27																				true	
25										false											
28																25.8					
29																	Overweight				
30,31,32,33,34,35,36,37,38			print		print + 'm'		print + 'kg'				print	print	print	print		print	print				
39										false											
40															n						
41																					true
39										false											
8										false											

Desk check table – Second data set

	valid Name	valid IC	gender	s_ height	height	s_ weight	weight	height Numeric	weight Numeric	DO WHI LE cond ition	name	identity CardN o	dob	blood Type	decision	bmi	weight Status	valid Height	valid Weight	valid Blood Type	valid Decision
1,2,3,4,5,6,7	false	false			0		0	true	true												
8										false											
9										true											
10											Noor Arzila										
11	true																				
9										false											
12										true											
13												75032 60706 14									
14		true																			
12										false											
15													26/03/ 75								
16			Femal e																		
17										false											
18				1.55																	
19					1.55																
20																		true			
17										false											
21										false											
22						48															
23							48.0														
24																			true		
21										false											
25										false											
26														AB-							
27																				true	
25										false											
28																20.0					
29																	Health y Weigh t print				
30,31,32,33 ,34,35,36,37,38			print		print + 'm'		print + 'kg'				print	print	print	print		print	print				
39										false											
40															y						
41																					true
39										false											
8										true											
9										false											
10																					
11	false																				
9										true											
10											45489										
11	false																				
9										true											
10											Ali										
11	true																				
9										false											
12										false											
13																					
14		false										bdbc									
12										true											
13												02070 80704 33									
14		true																			
12										false											
15													08/07/ 02								
16			Male																		
17										false											
18				1.28.9																	
19								false													
20								true										false			
17										true											
18				1.8																	
19					1.8																
20																		true			
17										false											

	valid Name	valid IC	gender	s_height	height	s_weight	weight	height Numeric	weight Numeric	DO WHI LE condition	name	identity CardNo	dob	blood Type	decision	bmi	weight Status	valid Height	valid Weight	valid Blood Type	valid Decision
21										false											
22						1000															
23							1000														
24																			false		
21										true											
22						ad															
23									false												
24									true										false		
21										true											
22						100															
23							100.0														
24																					
21										false											
25										false											
26																					
27														dads							
25										true											
26														A+							
27																					
25										false											
28																30.9					
29																	Obese				
30,31,32,33,34,35,36,37,38			print		print + 'm'		print + 'kg'				print	print	print	print		print	print				
39										false											
40															f						
41																					
39										true											false
40															N						
41																					true
39										false											
8										false											

2.0 Question 2

2.1 Java program code

```

package assignment;
import java.util.Scanner; //scanner class use to scan administrator's input
public class assignment1 {
    public static void main(String[]arg)
    {
        String name, identityCardNo, bloodType, weightStatus, dob, gender="";
        String s_height,s_weight; //store height and weight in string datatype in order to validate
        that administrator input is a valid number
        double height=0, weight=0;
        char decision; //administrator's decision to continue key in next patient's details or not
        boolean validName=false, validIC=false, validHeight, validWeight, validBloodType,
        validDecision; //for input validation
        boolean heightNumeric=true,weightNumeric=true; //for weight and height input validation
        (try catch)
        Scanner sc = new Scanner(System.in);
        do
        {
            System.out.println("Please enter the patient's details.");
        }
        do
        {
            System.out.print("Name: ");
            name = sc.nextLine();

```

```

    if (name.isBlank()==true) //check if the name input is blank
    {
        System.out.print("Invalid name, it should not be blank.\nPlease input again.\n");
        validName=false;
    }
    else
    {
        for (int i=0;i<name.length();i++) //checking if the name only contain space and
alphabet
        {
            char character = name.charAt(i);
            if (!Character.isLetter(character) && !Character.isWhitespace(character))
            {
                System.out.print("Invalid name, it should not contain special characters or
numbers.\nPlease input again.\n");
                validName=false;
                break;
            }
            else
            {validName=true;}
        }
    }

```

```

    }while(validName==false);
    do
    {
        System.out.print("Identity card number(no dash needed): ");
        identityCardNo = sc.nextLine();
        if(identityCardNo.length()!=12) //ic validation, ic must contain 12 digits
        {
            System.out.print("Invalid IC, it should contain 12 digits.\nPlease input again.\n");
            validIC=false;
        }
        else
        {
            for (int i=0;i<12;i++) //checking if the identityCardNo only contain numbers
            {
                char chara = identityCardNo.charAt(i);
                if (!Character.isDigit(chara))
                {
                    System.out.print("Invalid IC, it should contain only numbers.\nPlease input
again.\n");
                    validIC=false;
                    break;
                }
                else
                {validIC=true;}
            }
        }
    }
    }while(validIC==false);

```

to make it more accurate, you may consider to check if the date is correct or not. For example, if user keys in 883456789012, the birthdate will be 88/34/56. Which 88 and 34 is not a valid day and month.

```

        dob =identityCardNo.substring(4,6)+"/"
            +identityCardNo.substring(2,4)+"/"
            +identityCardNo.substring(0,2);    //store dob in format of DD/MM/YY
        if(identityCardNo.charAt(11)%2!=0)    //last digit even number is female, odd number is
male
    {
        gender = "Male";
    }
    else
    {
        gender = "Female";
    }
    do
    {
        System.out.print("Height(in meter): ");
        s_height = sc.nextLine();
        try //check if the height input by administrator is number
        {
            height = Double.parseDouble(s_height); //convert string to numeric type
        }
        catch (NumberFormatException e) //indicate the string is not in numeric format
        {
            heightNumeric=false;
        }
        if(heightNumeric==true) //validate true height value which should not be negative or zero or
higher than 3m.
        {
            if (height>0 && height<=3)
            {
                validHeight=true;
            }
            else
            {
                System.out.print("Invalid height, it should not be negative value or higher than
3m.\nPlease input again.\n");
                validHeight=false;
            }
        }
    }
    else
    {
        System.out.print("Invalid height, please enter a valid height which is a number.\n");
        validHeight=false;
        heightNumeric=true; //reset heightNumeric to true for try catch
    }
}while(validHeight==false);
do
{
    System.out.print("Weight(in kilogram): ");
    s_weight = sc.nextLine();
    try //check if the weight input by administrator is number
    {

```

```
weight = Double.parseDouble(s_weight); //convert string to numeric type
}
catch (NumberFormatException e) //indicate the string is not in numeric format
{
    weightNumeric=false;
}
if(weightNumeric==true)
{
    if (weight>0 && weight<=640 ) //validate true weight value which should not be negative
or zero or heavier than 640kg.
    {
        validWeight=true;
    }
    else
    {
        System.out.print("Invalid weight, it should not be 0 or lower than 0 or heavier than
640kg.\nPlease input again.\n");
        validWeight=false;
    }
}
else
{
    System.out.print("Invalid weight, please enter a valid height which is a number.\n");
    validWeight=false;
    weightNumeric=true; //reset weightNumeric to true for try catch
}
}while(validWeight==false);
do
{
    System.out.print("Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-): ");
    bloodType = sc.nextLine();
    if(bloodType.equals("A+") || bloodType.equals("A-")||
        bloodType.equals("B+")|| bloodType.equals("B-")||
        bloodType.equals("O+")|| bloodType.equals("O-")||
        bloodType.equals("AB+")|| bloodType.equals("AB-")) //validate blood type input
    {
        validBloodType=true;
    }
    else
    {
        System.out.print("Invalid blood type.\nPlease input again.\n");
        validBloodType=false;
    }
}while((validBloodType==false));
double bmi=weight/(Math.pow(height,2)); //calculate bmi value
if(bmi>=30) //assigning weight status
{
    weightStatus="Obese";
}
else if(bmi<30 && bmi>=25)
{
```

```

        weightStatus="Overweight";
    }
    else if(bmi<25 && bmi>=18.5)
    {
        weightStatus="Healthy Weight";
    }
    else
    {
        weightStatus="Underweight";
    }
    System.out.println("====="); //print
out patient's information
    System.out.println("Name: \t\t " + name);
    System.out.println("Identity card number: " + identityCardNo);
    System.out.printf("Height: \t\t %.2f m", height);
    System.out.printf("\nWeight: \t\t %.1f kg", weight);
    System.out.println("\nBlood type: \t\t " + bloodType);
    System.out.println("Date of birth[DD/MM/YY]: " + dob);
    System.out.println("Gender: \t\t " + gender);
    System.out.printf("BMI: \t\t %.1f", bmi);
    System.out.println("\nWeight status: \t\t " + weightStatus);
    System.out.println("=====");
do
{
    System.out.print("Continue for the next patient's details?[Y/N]: "); //asking if administrator
want to continue for the next patient's details
    decision = sc.next().charAt(0);
    sc.nextLine(); // Consume newline left-over
    if(decision!='Y' && decision!='y' && decision!='N' && decision!='n')
    {
        System.out.println("Invalid input, please key in either Y or N.");
        validDecision=false;
    }
    else
    {
        validDecision=true;
    }
}while(validDecision==false);
}while(decision=='Y' || decision=='y');
}
}

```

2.2 Description

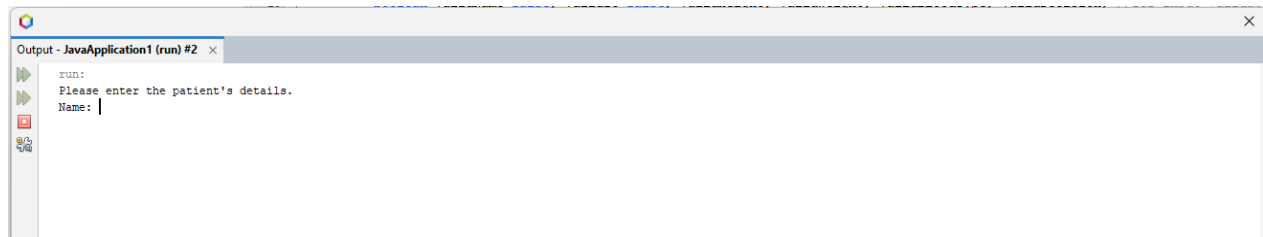


Figure 1.1 Prompt for name

This program is built to enable administrator to key in the patient's details (name, identity card number, height, weight and blood type), after that, the program will print out the patient's information (name, identity card number, height, weight, blood type, date of birth, gender, BMI, and weight status), thus helping the administrator to handle patients' records.

After running the program, the program will prompt for the patient's name as shown in Figure 1.1.

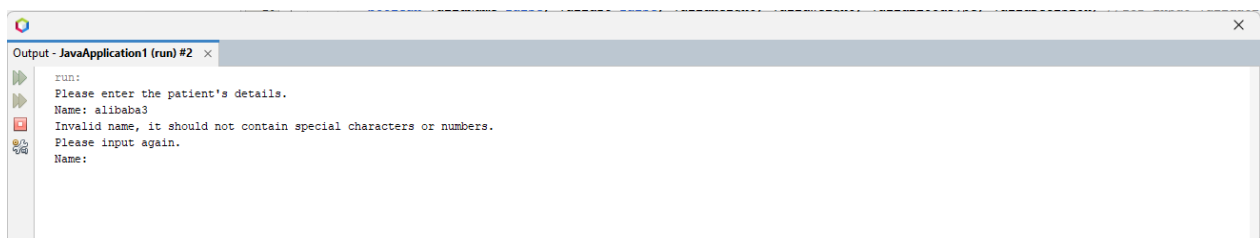


Figure 1.2 Invalid name – contain numbers

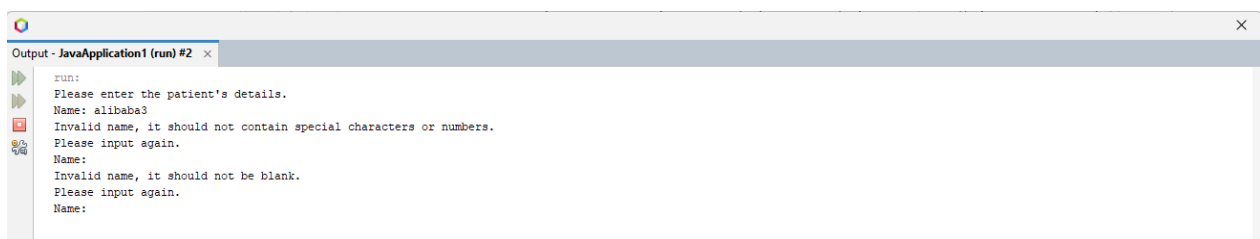


Figure 1.3 Invalid name – blank



Figure 1.4 Invalid name – contain special characters

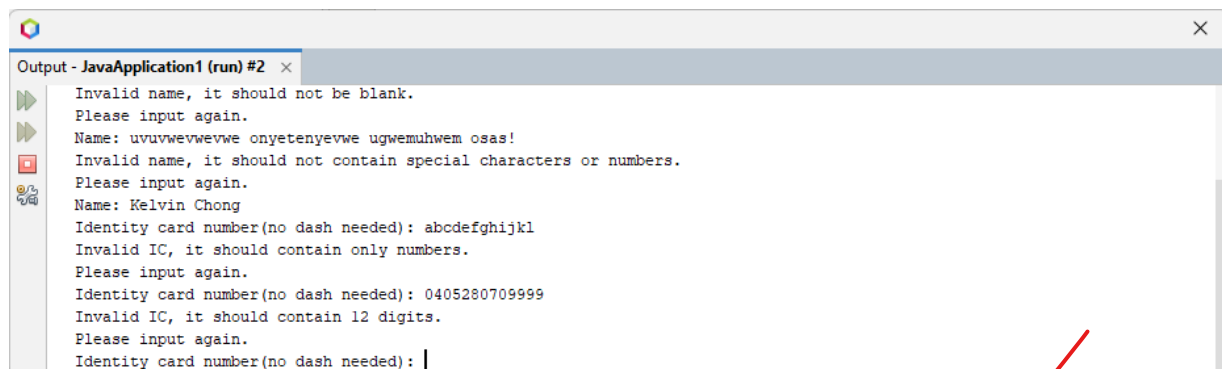
The program will validate name input by the administrator, if the name input is blank or contain numbers and special characters, the program will prompt error message and allow the administrator to repeat the input for patient's name until the input is a valid name. (Figure 1.2, Figure 1.3, Figure 1.4)



```
Output - JavaApplication1 (run) #2 x
run:
Please enter the patient's details.
Name: alibaba3
Invalid name, it should not contain special characters or numbers.
Please input again.
Name:
Invalid name, it should not be blank.
Please input again.
Name: uvuvvewvewve onyeteneyevwe ugwemuhwem osas!
Invalid name, it should not contain special characters or numbers.
Please input again.
Name: Kelvin Chong
Identity card number(no dash needed):
```

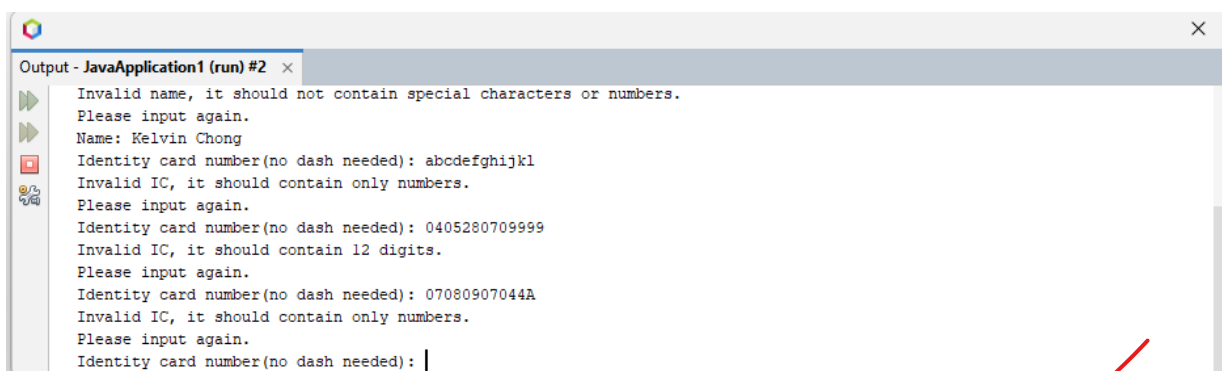
Figure 2.1 Prompt for identity card number

After input a valid name, the program will continue prompt for identity card number as shown in Figure 2.1.



```
Output - JavaApplication1 (run) #2 x
Invalid name, it should not be blank.
Please input again.
Name: uvuvvewvewve onyeteneyevwe ugwemuhwem osas!
Invalid name, it should not contain special characters or numbers.
Please input again.
Name: Kelvin Chong
Identity card number(no dash needed): abcdefghijkl
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): 0405280709999
Invalid IC, it should contain 12 digits.
Please input again.
Identity card number(no dash needed): |
```

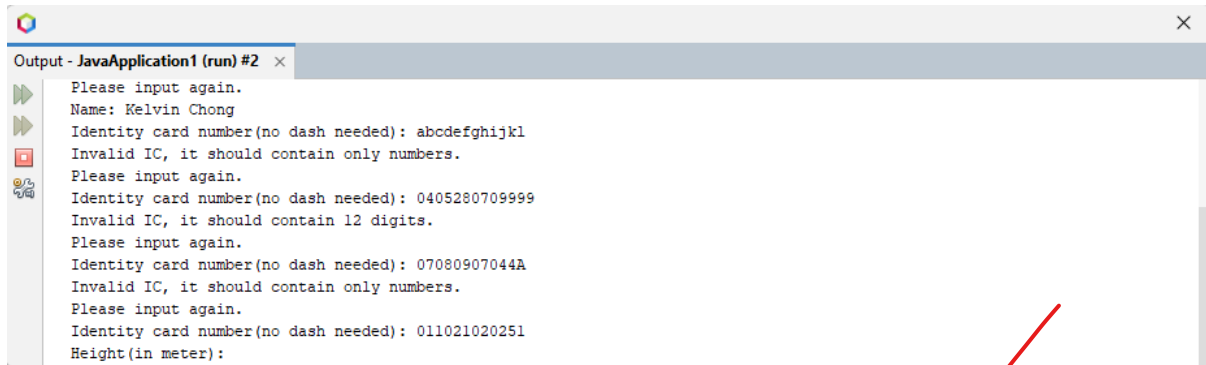
Figure 2.2 Invalid identity card number – not made up of 12 digits



```
Output - JavaApplication1 (run) #2 x
Invalid name, it should not contain special characters or numbers.
Please input again.
Name: Kelvin Chong
Identity card number(no dash needed): abcdefghijkl
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): 0405280709999
Invalid IC, it should contain 12 digits.
Please input again.
Identity card number(no dash needed): 07080907044A
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): |
```

Figure 2.3 Invalid identity card number – contain non-numeric character

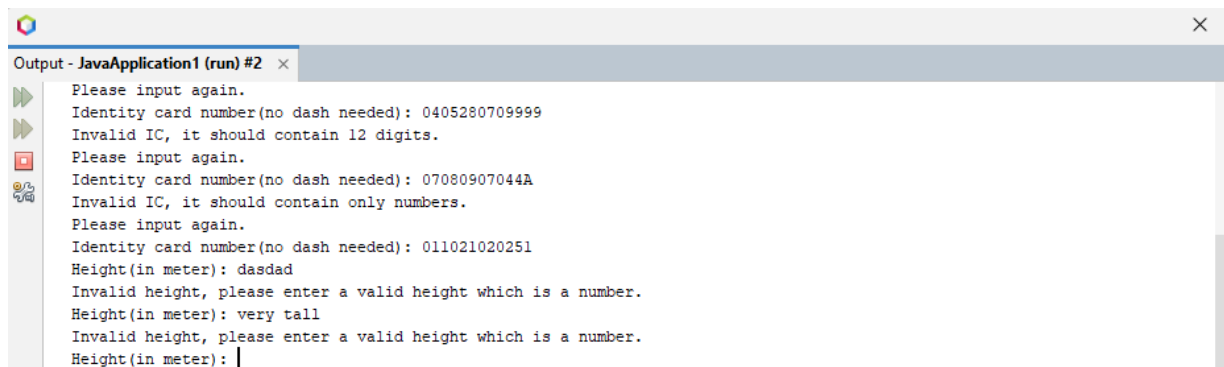
The program will validate identity card number input by the administrator which should be exactly 12 numbers and does not include non-numeric character. The program will prompt error message and allow the administrator to repeat the input for patient's identity card number for every invalid input. (Figure 2.2, Figure 2.3)



```
Output - JavaApplication1 (run) #2 x
Please input again.
Name: Kelvin Chong
Identity card number(no dash needed): abcdefghijk1
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): 0405280709999
Invalid IC, it should contain 12 digits.
Please input again.
Identity card number(no dash needed): 07080907044A
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): 011021020251
Height(in meter):
```

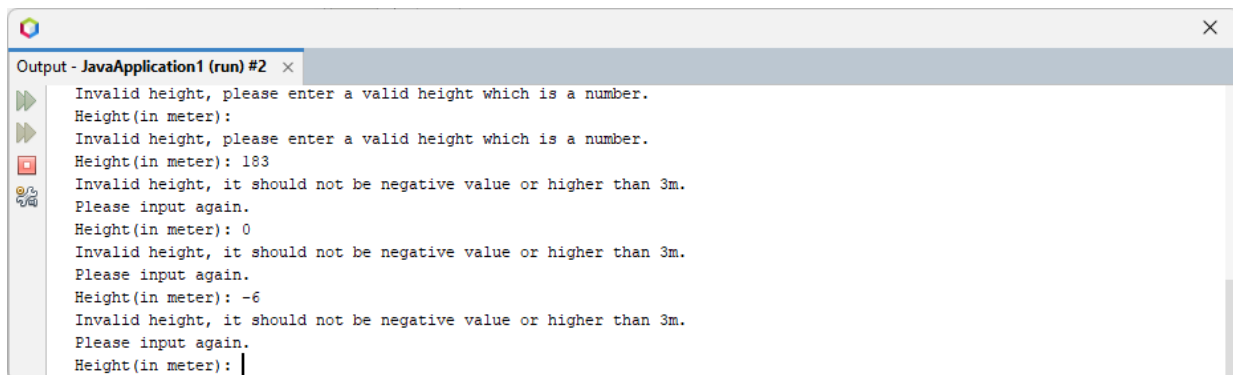
Figure 3.1 Prompt for patient's height

After administrator input a valid identity card number, the program will ask for patient's height in meter as shown in Figure 3.1.



```
Output - JavaApplication1 (run) #2 x
Please input again.
Identity card number(no dash needed): 0405280709999
Invalid IC, it should contain 12 digits.
Please input again.
Identity card number(no dash needed): 07080907044A
Invalid IC, it should contain only numbers.
Please input again.
Identity card number(no dash needed): 011021020251
Height(in meter): dasdad
Invalid height, please enter a valid height which is a number.
Height(in meter): very tall
Invalid height, please enter a valid height which is a number.
Height(in meter):
```

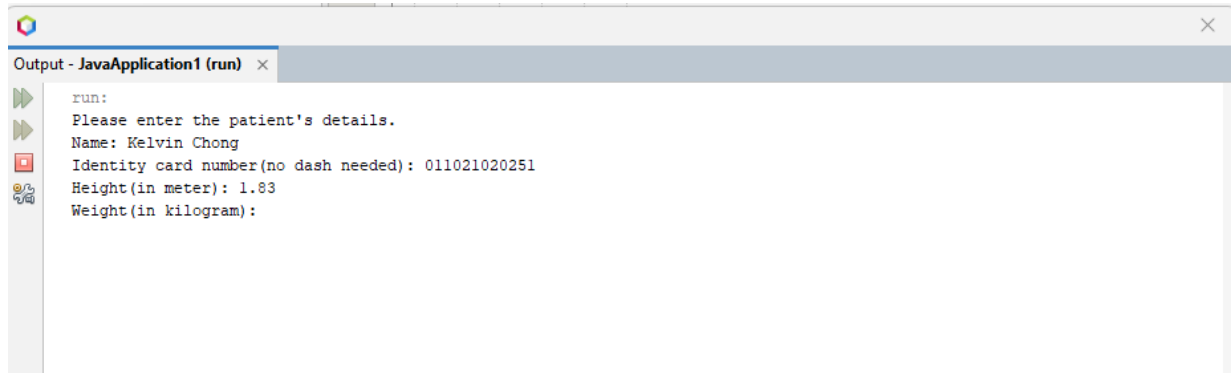
Figure 3.2 Invalid height – contain non-numeric character



```
Output - JavaApplication1 (run) #2 x
Invalid height, please enter a valid height which is a number.
Height(in meter):
Invalid height, please enter a valid height which is a number.
Height(in meter): 183
Invalid height, it should not be negative value or higher than 3m.
Please input again.
Height(in meter): 0
Invalid height, it should not be negative value or higher than 3m.
Please input again.
Height(in meter): -6
Invalid height, it should not be negative value or higher than 3m.
Please input again.
Height(in meter):
```

Figure 3.3 Invalid height – not within normal range (0m<x<= 3m)

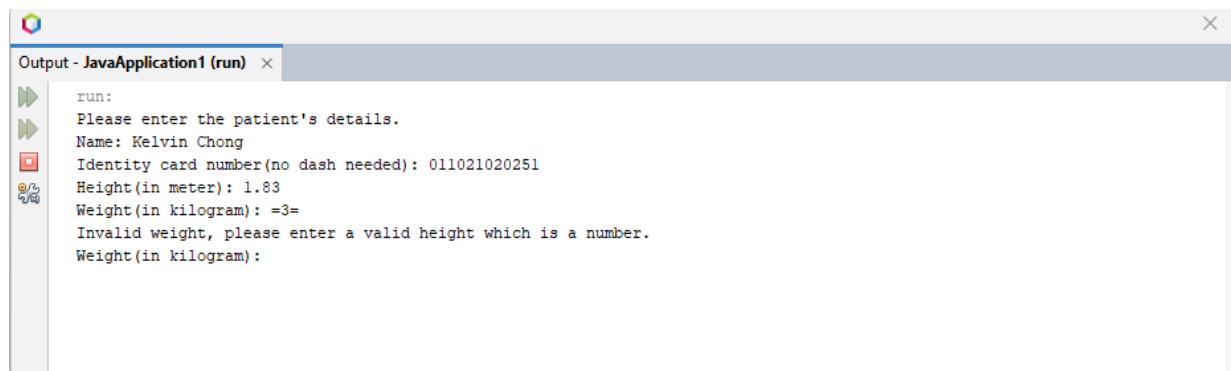
The program will validate the patient's height input by the administrator which should be numeric and within normal height range (0m<x<= 3m). The program will prompt error message and allow the administrator to repeat the input for patient's height for every invalid input. (Figure 3.2, Figure 3.3)



```
run:
Please enter the patient's details.
Name: Kelvin Chong
Identity card number(no dash needed): 011021020251
Height(in meter): 1.83
Weight(in kilogram):
```

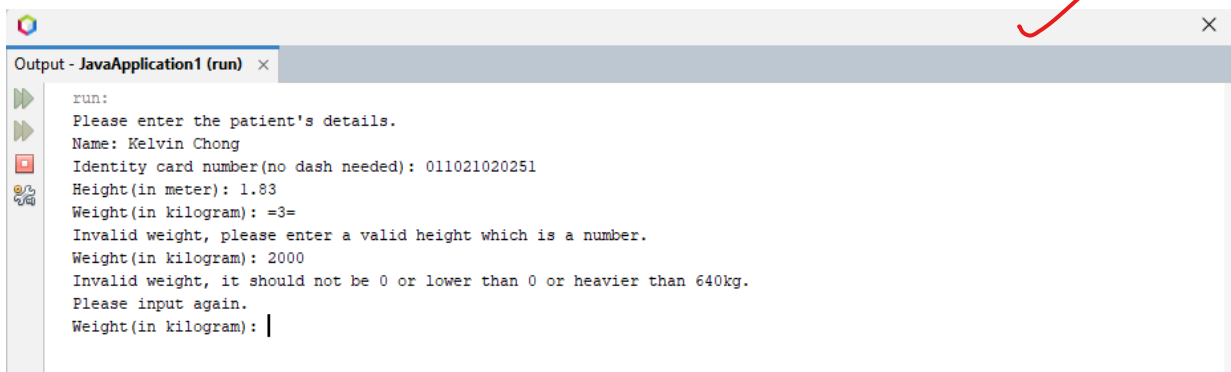
Figure 4.1 Prompt for patient's weight

After administrator input a valid height, the program will ask for patient's weight in kilogram as shown in Figure 4.1.



```
run:
Please enter the patient's details.
Name: Kelvin Chong
Identity card number(no dash needed): 011021020251
Height(in meter): 1.83
Weight(in kilogram): =3=
Invalid weight, please enter a valid height which is a number.
Weight(in kilogram):
```

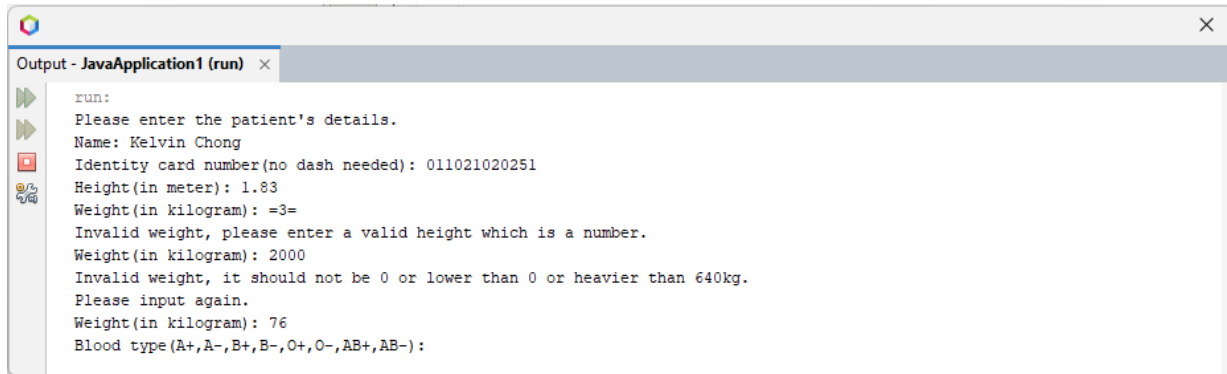
Figure 4.2 Invalid weight - contain non-numeric character



```
run:
Please enter the patient's details.
Name: Kelvin Chong
Identity card number(no dash needed): 011021020251
Height(in meter): 1.83
Weight(in kilogram): =3=
Invalid weight, please enter a valid height which is a number.
Weight(in kilogram): 2000
Invalid weight, it should not be 0 or lower than 0 or heavier than 640kg.
Please input again.
Weight(in kilogram): |
```

Figure 4.3 Invalid weight - not within normal range ($0\text{kg} < x \leq 640\text{kg}$)

The program will validate the patient's weight input by the administrator which should be numeric and within normal weight range ($0\text{kg} < x \leq 640\text{kg}$). The program will prompt error message and allow the administrator to repeat the input for patient's weight for every invalid input. (Figure 4.2, Figure 4.3)

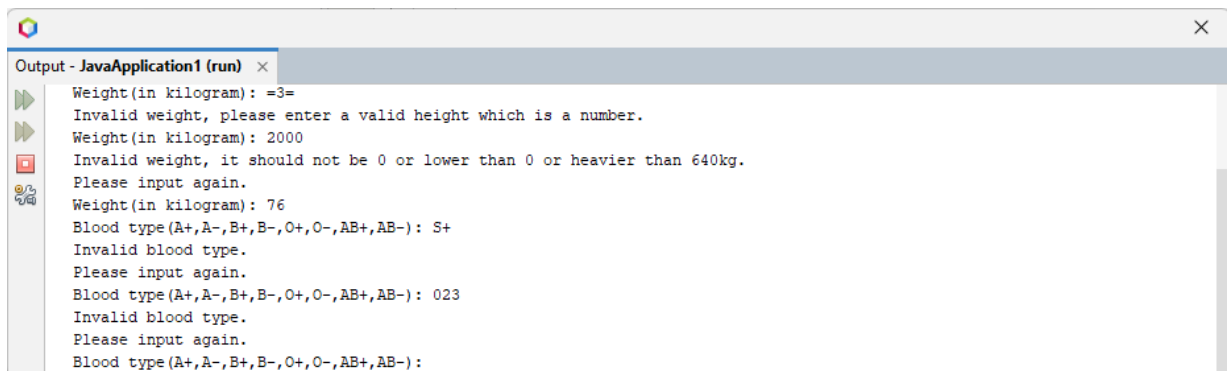


```

run:
Please enter the patient's details.
Name: Kelvin Chong
Identity card number(no dash needed): 011021020251
Height(in meter): 1.83
Weight(in kilogram): =3=
Invalid weight, please enter a valid height which is a number.
Weight(in kilogram): 2000
Invalid weight, it should not be 0 or lower than 0 or heavier than 640kg.
Please input again.
Weight(in kilogram): 76
Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-):
  
```

Figure 5.1 Prompt for patient's weight

After administrator input a valid weight, the program will ask for patient's blood type with the given list (A+,A-,B+,B-,O+,O-,AB+,AB-) as shown in Figure 5.1.

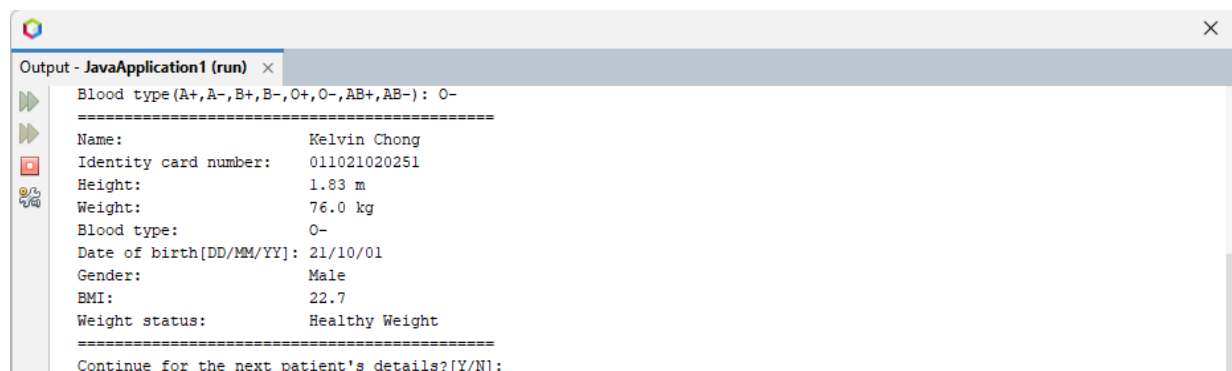


```

Weight(in kilogram): =3=
Invalid weight, please enter a valid height which is a number.
Weight(in kilogram): 2000
Invalid weight, it should not be 0 or lower than 0 or heavier than 640kg.
Please input again.
Weight(in kilogram): 76
Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-): S+
Invalid blood type.
Please input again.
Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-): 023
Invalid blood type.
Please input again.
Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-):
  
```

Figure 5.2 Invalid blood type – not in given list

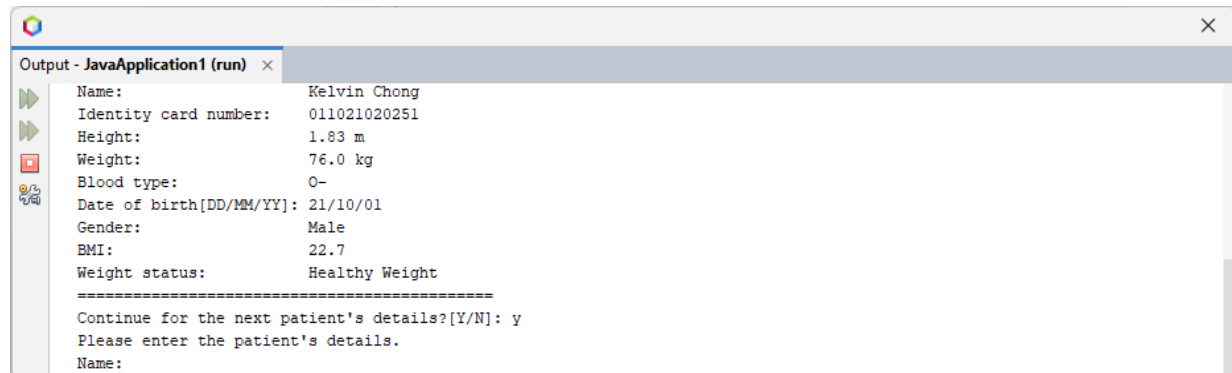
The program will validate the patient's blood type input by the administrator which should be in the given list (A+,A-,B+,B-,O+,O-,AB+,AB-). The program will prompt error message and allow the administrator to repeat the input for patient's blood type for every invalid input. (Figure 5.2)



```

Blood type(A+,A-,B+,B-,O+,O-,AB+,AB-): O-
=====
Name: Kelvin Chong
Identity card number: 011021020251
Height: 1.83 m
Weight: 76.0 kg
Blood type: O-
Date of birth[DD/MM/YY]: 21/10/01
Gender: Male
BMI: 22.7
Weight status: Healthy Weight
=====
Continue for the next patient's details?[Y/N]:
  
```

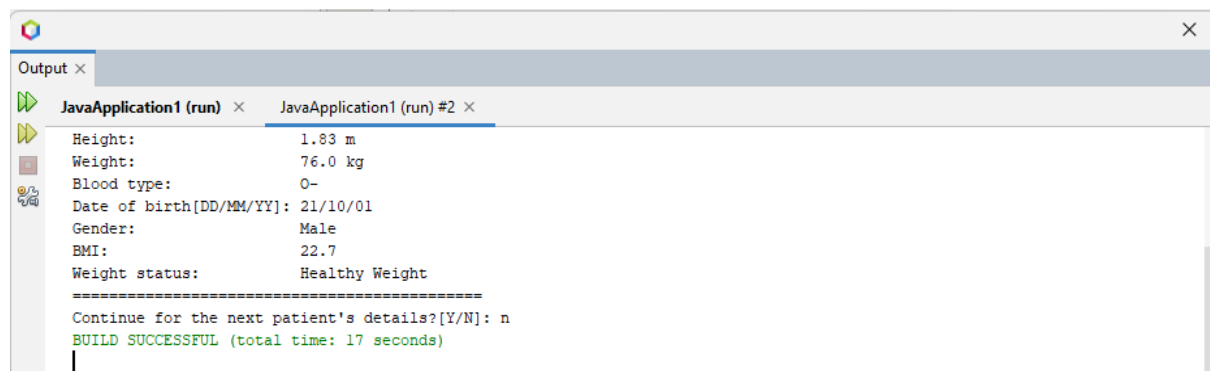
Figure 6.1 Display patient's information and ask for decision



```
Output - JavaApplication1 (run) x
Name: Kelvin Chong
Identity card number: 011021020251
Height: 1.83 m
Weight: 76.0 kg
Blood type: O-
Date of birth[DD/MM/YY]: 21/10/01
Gender: Male
BMI: 22.7
Weight status: Healthy Weight
=====
Continue for the next patient's details?[Y/N]: y
Please enter the patient's details.
Name:
```

Figure 6.2 Continue for next patient's details

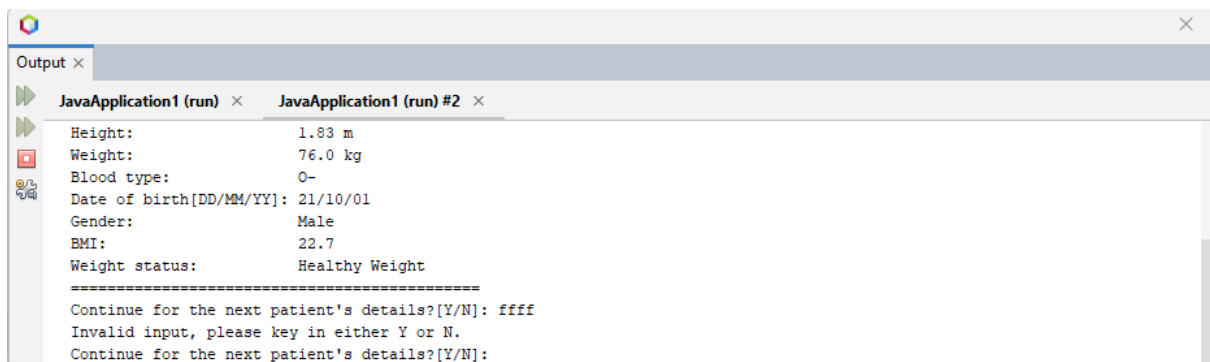
After administrator input a valid blood type, the program will display patient's information (name, identity card number, height, weight, blood type, date of birth, gender, BMI, and weight status) as shown in Figure 6.1. After that, the program will ask for administrator's decision to either continue handling the next patient's details or close the program. If the administrator key in either 'Y' or 'y' or 'yes', the program will loop again for the administrator to key in patient's details as shown in Figure 6.2.



```
Output x
JavaApplication1 (run) x JavaApplication1 (run) #2 x
Height: 1.83 m
Weight: 76.0 kg
Blood type: O-
Date of birth[DD/MM/YY]: 21/10/01
Gender: Male
BMI: 22.7
Weight status: Healthy Weight
=====
Continue for the next patient's details?[Y/N]: n
BUILD SUCCESSFUL (total time: 17 seconds)
```

Figure 6.3 End the program

If the administrator choose to shut down the program and key in 'n' or 'N' or 'no', the program will eventually end as shown in Figure 6.3.



```
Output x
JavaApplication1 (run) x JavaApplication1 (run) #2 x
Height: 1.83 m
Weight: 76.0 kg
Blood type: O-
Date of birth[DD/MM/YY]: 21/10/01
Gender: Male
BMI: 22.7
Weight status: Healthy Weight
=====
Continue for the next patient's details?[Y/N]: ffff
Invalid input, please key in either Y or N.
Continue for the next patient's details?[Y/N]:
```

Figure 6.4 Invalid decision

The program will also validate the decision input by the administrator which should be either 'y' or 'Y' or 'n' or 'N' or 'yes' or 'no'. The program will prompt error message and allow the administrator to repeat the input for the decision for every invalid input. (*Figure 6.4*)

