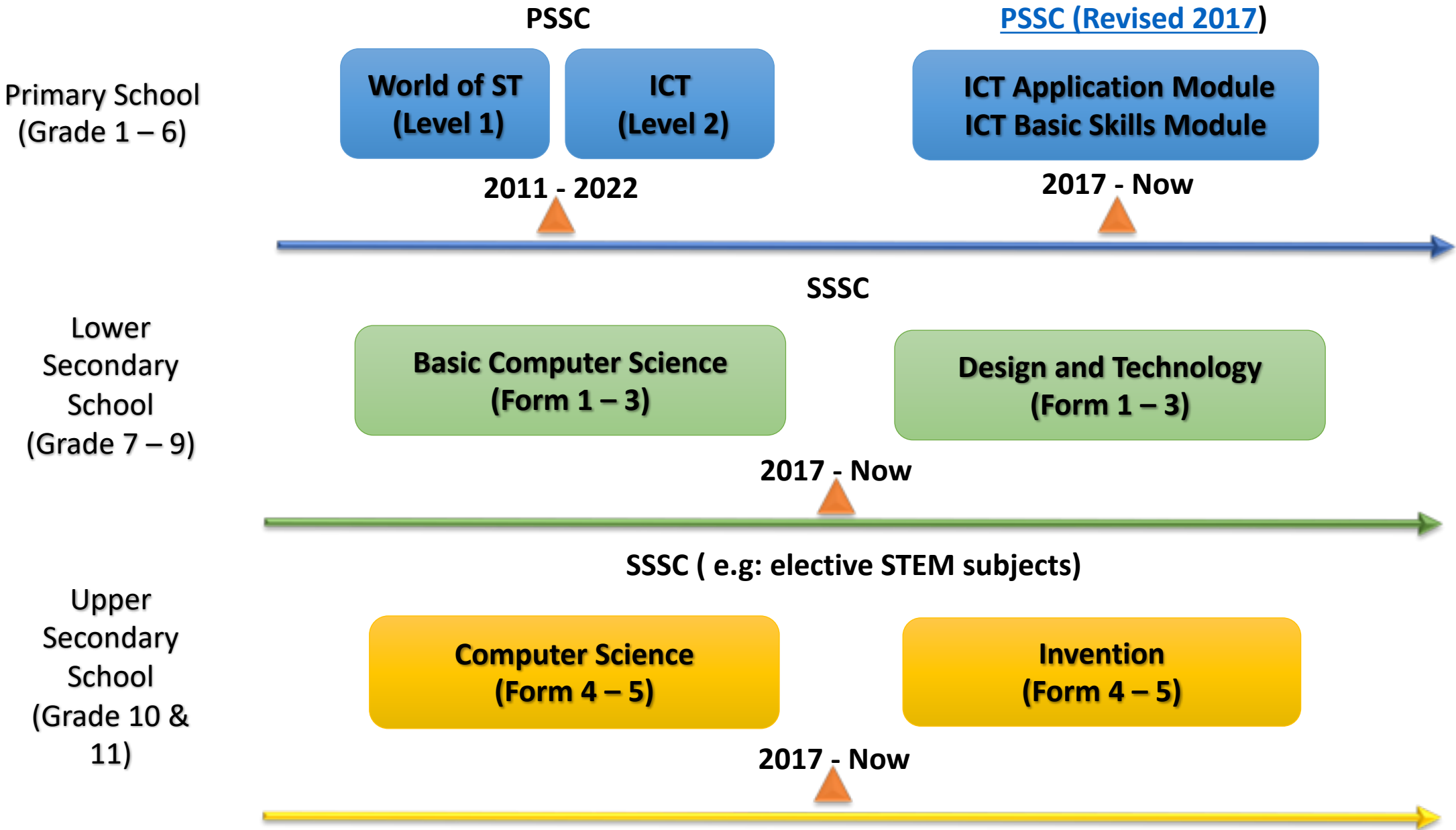


COMPUTATIONAL THINKING IN PRIMARY AND SECONDARY SCHOOL CURRICULUM

CURRICULUM DEVELOPMENT DIVISION

Ministry of Education Malaysia

ICT (Computing) Curriculum in Malaysia Primary and Secondary School



Supporting materials for implementing computing in schools

ICT Basic Skills Module



Learning Modules for Basic Computer Science (ASK)



ICT Application Module



Learning Modules for Computer Science (SK)



Content Standard of Lower Secondary BCS and D&T

<p>Basic Computer Science - Lower Secondary</p>	<p>Form 1: 1.0 Basic Concepts of Computational Thinking (Decomposition, Pattern Recognition, Abstraction and Generalization) 2.0 Data Representation (Data Size) 3.0 Algorithms (Selection & Repetition control structure) 4.0 Instructions Code (SCRATCH and HTML)</p> <p>Form 2: 1.0 Basic Concepts of Computational Thinking 2.0 Data Representation (Number system) 3.0 Algorithms (Sequence control structure, selection & iteration nesting) 4.0 Instruction Code (Python)</p> <p>Form 3: 1.0 Basic Concepts of Computational Thinking (Program Development Project) 2.0 Data Representation (Cryptography In Data Security) 3.0 Algorithms (Sort and Search) 4.0 Command Codes (Database and SQL)</p>
<p>Design and Technology - Lower Secondary</p>	<p>Form 2 - 1.0 Inventive Problem Solving 2.0 Applications Of Technology 2.1 Manufacturing Technology 2.4 Electronic Design: Involves programming such as SCRATCH, MicroBit, Arduino etc. as well as the use of micro-controllers to controls devices (robotic elements and IoT)</p> <p>Form 3 - 1.0 Applications Of Technology 2.0 Product Development 3.0 The Role Of Design In Business</p> <p style="margin-left: 400px;">} Mechatronic technology & Product Making: Involves programming such as SCRATCH, MicroBit, Arduino etc. as well as the use of micro controllers to controls devices (robotic and IoT elements)</p>

Project Work

Form 3 SSSC Basic Computer Science Project Guide



Phases in developing the application:

- (i) Analyze the problem
- (ii) Program design
- (iii) Coding
- (iv) Testing and debugging
- (v) Documentation

A combination of contents that has been learned and integrated at least with another subject

1. Documentation
2. Application
3. Presentation

6 Dimension:

1. Information Gathering
2. Planning
3. Applying
4. Reflection
5. Communication
6. Ethical & Spritual

Example of other subjects contents

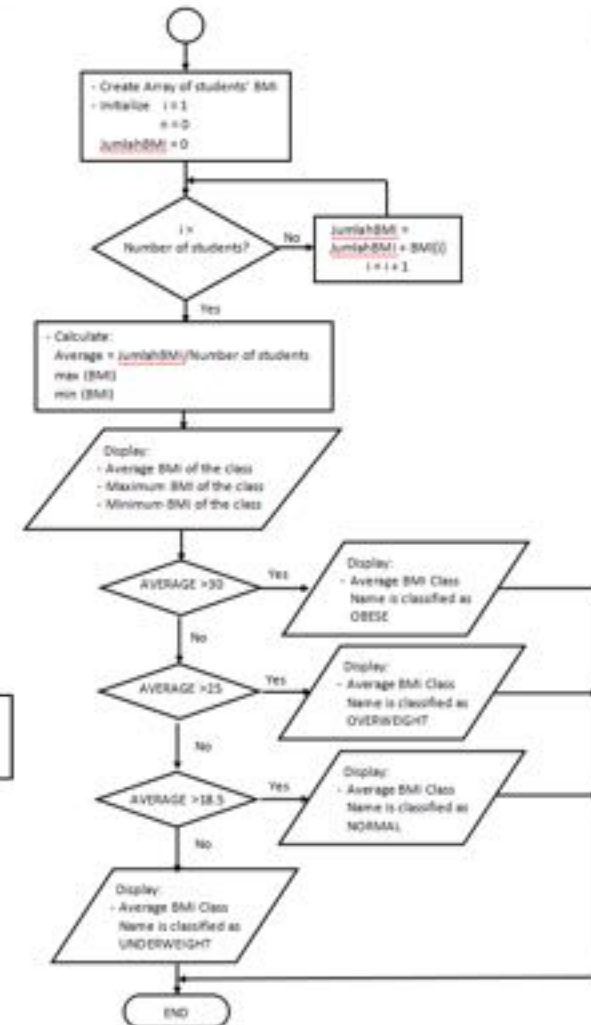
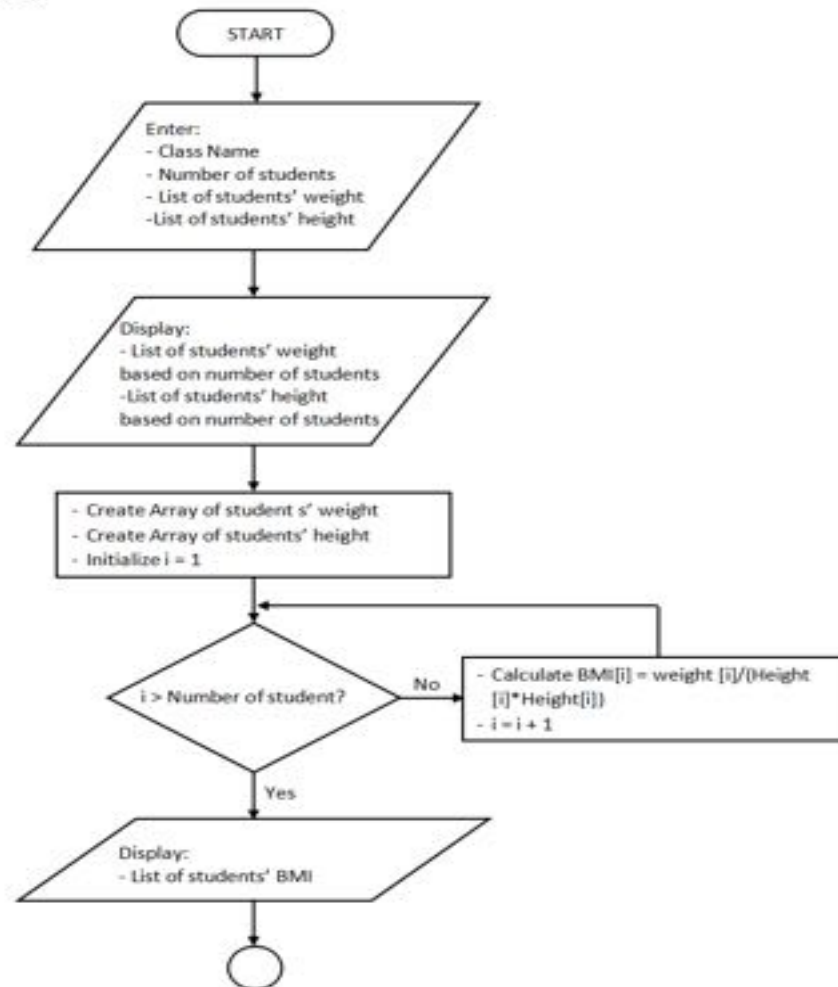
SSSC Science Form 2
Project Based Learning
3.0 Nutrition: Obesity
4.0 Human Health: Disease
5.0 Water & Solution: Water Auditing

SSSC Physical & Health Education Form 2
6.0 Disease: How to prevent & reduce risk factors
8.0 Nutrition: Practice healthy & safe eating

Task 1: Individual Health Advice

Task 2: Reviewing One-Class Health Levels

Task 2:



Coding

```
# Calculate BMI from list of Data

# Function to calculate BMI
def KiraBMI (Brt, Tgi):
    BMI = int(Brt/(Tgi*Tgi))
    return BMI

# Input Data from a class : Class name, Number of students, weight and height of each student.
Kelas = input ("\nEnter the class name : ")
BilMurid = int (input ("\nEnter the number of students in the class : "))
NamaMurid = list(map(str,input("\nEnter the first name of each student (separate with a space): ").strip().split()))[:BilMurid]
BeratMurid = list(map(float,input("\nEnter the weight(kg)of each student (separate with a space): ").strip().split()))[:BilMurid]
TinggiMurid = list(map(float,input("\nEnter the height(m)of each student (separate with a space): ").strip().split()))[:BilMurid]

#List all the data entered
print("\nList of students' Name = ", NamaMurid)
print("List of students' weight = ", BeratMurid)
print("List of students' height = ", TinggiMurid)
print(" ")

# Build array of students' name, weight and height
Nama = []
Nama = NamaMurid
Berat = []
Berat = BeratMurid
#print (Berat)
Tinggi = []
Tinggi = TinggiMurid
#print (Tinggi)

#call function to calculate BMI in KiraBMI as an array
SenaraiBMI = []
SenaraiBMI = [KiraBMI(Berat[i],Tinggi[i]) for i in range(len(Berat))]

#list of students with BMI
NoOfObese = 0
NoOfOverweight = 0
NoOfNormal = 0
NoOfUnderweight = 0
count = 0
while count < (BilMurid):
    if SenaraiBMI[count] > 30:
        Obese = SenaraiBMI[count]
        NamaObese = NamaMurid[count]
        print (NamaObese,"(",Obese,")is obese")
        NoOfObese = NoOfObese+1
    else:
        if SenaraiBMI[count] > 25:
            Overweight = SenaraiBMI[count]
            NamaOverweight = NamaMurid[count]
            print (NamaOverweight,"(",Overweight,")is overweight")
            NoOfOverweight = NoOfOverweight+1
        else:
            if SenaraiBMI[count] > 18.5:
                Normalweight = SenaraiBMI[count]
                NamaNormal = NamaMurid[count]
                print (NamaNormal,"(",Normalweight,")is Normal")
                NoOfNormal = NoOfNormal+1
            else:
                Underweight = SenaraiBMI[count]
                NamaUnderweight = NamaMurid[count]
                print (NamaUnderweight,"(",Underweight,")is underweight")
                NoOfUnderweight = NoOfUnderweight+1
            count = count + 1

#Analysis for BMI
print ("-----")
print ("\nANALYSIS OF BMI FOR CLASS",Kelas)
sum = 0
for Jumlah in SenaraiBMI:
    sum = sum + Jumlah
PurataBMI = sum / len(SenaraiBMI)
print ("\nAverage BMI in class",Kelas,"is ", int(PurataBMI))
print ("Maximum BMI in the class is ", max(SenaraiBMI))
print ("Mininum BMI in the class is ", min(SenaraiBMI))

if PurataBMI > 30:
    print ("\nAverage BMI for",Kelas,"is classified as OBESE")
else:
    if PurataBMI > 25:
        print ("\nAverage BMI for",Kelas,"is classified as UNDERWEIGHT")

PercentNormal = (NoOfNormal/BilMurid)*100
PercentOverweight = (NoOfOverweight/BilMurid)*100
PercentObese = (NoOfObese/BilMurid)*100
PercentUnderweight = (NoOfUnderweight/BilMurid)*100
print ("\nNumber of Students with NORMAL BMI = ", NoOfNormal," , Percentage = ",PercentNormal,"%")
print ("Number of Students with OVERWEIGHT BMI = ", NoOfOverweight," , Percentage = ",PercentOverweight,"%")
print ("Number of Students with OBESE BMI = ", NoOfObese," , Percentage = ",PercentObese,"%")
print ("Number of Students with UNDERWEIGHT BMI = ", NoOfUnderweight," , Percentage = ",PercentUnderweight,"%")
```

Output

```
>>>
===== RESTART: D:\Projek Sains\CallFromListdata.py =====

Enter the class name : 5J

Enter the number of students in the class : 3

Enter the first name of each student (separate with a space): smith lee matt

Enter the weight(kg)of each student (separate with a space): 56 50 67

Enter the height(m)of each student (separate with a space)): 1.52 1.6 1.7

List of students' Name = ['smith', 'lee', 'matt']
List of students' weight = [56.0, 50.0, 67.0]
List of students' height = [1.52, 1.6, 1.7]

smith ( 24 )is Normal
lee ( 19 )is Normal
matt ( 23 )is Normal
-----

ANALYSIS OF BMI FOR CLASS 5J

Average BMI in class 5J is 22
Maximum BMI in the class is 24
Minumum BMI in the class is 19

Average BMI for 5J is classified as NORMAL

Number of Students with NORMAL BMI = 3 , Percentage = 100.0 %
Number of Students with OVERWEIGHT BMI = 0 , Percentage = 0.0 %
Number of Students with OBESE BMI = 0 , Percentage = 0.0 %
Number of Students with UNDERWEIGHT BMI = 0 , Percentage = 0.0 %
|>>>
===== RESTART: D:\Projek Sains\BMI.py =====
```


Photos of Students showing their skill in coding





What is CT

It's about how human think. It is a thinking skills. It is NOT think like computer because computer DOES NOT think

Thank You

4 Elements of ICT Skills in Primary School

1. Deliver creative and innovative ideas and information through ICT
2. Search, collect, process and use information through ICT
3. Using Computational Thinking for problem solving
 - a. Analyze data through pattern recognition
 - b. Present data or ideas logically and systematically through graphs, charts or images
 - c. Decompose complex situation, condition or problem in the form of small components
 - d. Determine important aspects of a situation, condition or problem
 - e. Apply a step-by-step approach (algorithm) systematically in a given situation, condition or problem
 - f. Interpret solution visually through models, simulations or flow charts.
 - g. Identify and analyze the appropriate ICT resources to achieve the best possible solution effectively using computer technology.
4. Practice ethical and responsible attitude in ICT

