**iCompute**

**Supervisor:** Krish Narayanan / Preethi Narayanan

**Grades:** 4th and 5th only

**Participants:** 2

**Description:** Participants will be tested on the following.
- Knowledge of basic computer concepts, including hardware and software
- Understanding of how computers solve problems
- Ability to create a simple program using a graphical interface

**Duration:** 45 minutes

**Sections:** 3

The first two sections do not require the use of a computer. The last section should be completed with a computer. The following describes each section with the knowledge area italicized.

1. Multiple-choice questions on *computing basics* - hardware, software, programming languages, networks, etc.
2. Short answer question(s) on *theoretical computing* - binary, algorithms, pseudocode, flowcharts, decomposition of problems, etc.
3. Exercise on *practical computing* – creation of a program for a given task.

**Rules:** The following rules should be followed at all times.
- Participants will not be expected to read or write code in a traditional programming language, such as, Java, Python, C++.
- Participants are not allowed to use the computer for the first two sections.
- Each team will have access to one computer.
- Participants are not allowed to use any other program or website during the competition.
- Each team will be provided with writing instruments.
- Participants are not allowed to bring any electronic devices.

**Scoring:** 50 points

**Breakdown**
- Section A - 5 multiple-choice questions, 2 points each
  - Each response will either get a score of 0 or 2.
- Section B - 2 short answer questions, 5 points each
  - The questions will be broken down into parts with individual points assigned. Each response will get a score based on whether it addresses all parts of the question.
- Section C - One programming activity using Scratch 2.0 offline editor, 30 points total, based on the following breakdown:
  - 10 points for a correct algorithm that uses proper programming constructs
  - 10 points for a working program
  - 5 points for attention to precision
  - 5 points for creativity and/or efficiency (tie-breaker)

**Website:** [http://emunix.emich.edu/~krish/iCompute/](http://emunix.emich.edu/~krish/iCompute/)
Resources needed:

- Hardware – PC or Mac, Laptop or Desktop
- Software – Scratch (free)
  - Online app can be found at http://www.scratch.mit.edu
  - Offline editor can be downloaded from https://scratch.mit.edu/scratch2download/

Sample questions:

Section A

1) What is the name of the unit that helps store data in a computer?
   a. CPU
   b. Input
   c. Memory
   d. Output

2) This provides a step-by-step procedure for performing a task.
   a. Keyboard
   b. Algorithm
   c. Internet
   d. Windows

3) Which one of the following is not a programming language?
   a. Java
   b. HTML
   c. C++
   d. Binary

Section B

1) Convert 234 from decimal to binary.
   a. Show your steps for conversion. (3 pts.)
   b. Write the final answer and circle it. (2 pts.)

2) The following is the algorithm for a certain task.
   i. Add age1 and age2. Store result in x.
   ii. Add age3 and age4. Store result in y.
   iii. Add age5 and age6. Store result in z.
   iv. Add x, y, and z. Store result in total.
   v. Print total divided by 6.
   a. Explain in a sentence or two what this algorithm is doing. What is the final result it is computing? (3 pts.)
   b. If age1=3, age2=5, age3=8, age4=5, age5=9, and age6=1, execute the 5 steps in this algorithm using these values. (2 pts.)

3) You are given a bag full of coins - pennies, nickels, dimes, and quarters. You have to sort them into four boxes, each holding one type of coin only. So, the first box should have all pennies, the second all nickels, and so on. Draw a flowchart for this activity. Keep in mind you can take only one coin at a time and place it in the appropriate box before you pick another one.

Section C

1) Using Scratch, create a program that makes an actor bounce around on the screen.
   The actor should start on the top, left corner of the screen and move to each corner in a circular fashion. The actor should spend at least 3 seconds in each corner. Finally, the actor should stop in the center and say “I’m done going around!”.
2) Using Scratch, create a program that says a joke. Use any two actors of your choice and make them act out the following joke. Include a backdrop for the stage and play music in the background.

Knock, knock.
Who's there?
Canoe.
Canoe who?
Canoe help me with my homework?

3) Using Scratch, create a program that will accept a value between 0 and 100. This value represents a temperature. The program should suggest an activity based on the temperature, per the following criteria:

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-39</td>
<td>Skiing</td>
</tr>
<tr>
<td>40-59</td>
<td>Golf</td>
</tr>
<tr>
<td>60-79</td>
<td>Tennis</td>
</tr>
<tr>
<td>80-100</td>
<td>Swimming</td>
</tr>
</tbody>
</table>

Coach’s workshop: Coaches will have the opportunity to learn the software (Scratch) from the supervisor and/or college students. At least two workshops will be scheduled. Coaches will also be referred to a number of resources on the web.

Resources:
1. [http://www.gcflearnfree.org/computers](http://www.gcflearnfree.org/computers) - a good intro to computers.
2. [https://code.org/learn](https://code.org/learn) – a complete resource for teaching kids programming in a fun, interactive manner.
4. [http://www.youtube.com/user/csunplugged](http://www.youtube.com/user/csunplugged) - Unplugged YouTube channel.
5. [http://www.youtube.com/watch?v=RWEAv1B1_qA](http://www.youtube.com/watch?v=RWEAv1B1_qA) - a lecture on teaching the principles of computer science to kids.
7. [https://www.google.com/edu/computational-thinking/](https://www.google.com/edu/computational-thinking/) - Google’s repository of K-12 resources.